AFRL-IF-RS-TR-2005-293
Final Technical Report
August 2005

SECURE WIRELESS FAULT TOLERANT TUNABLE NETWORKS (SWIFT)

University of California at Riverside

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. M099

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-293 has been reviewed and is approved for publication.

APPROVED: /s/

ALAN J. AKINS
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** AUGUST 2005	**3. REPORT TYPE AND DATES COVERED** Final Jul 01 – Sep 04

4. TITLE AND SUBTITLE
SECURE WIRELESS FAULT TOLERANT TUNABLE NETWORKS (SWIFT)

5. FUNDING NUMBERS
C - F30602-01-2-0536
PE - 62301E
PR - FTNP
TA - M0
WU - 99

6. AUTHOR(S)
Chinya V. Ravishankar, Srikanth Krishnamurthy,
Michalis Faloutsos and Satish Tripathi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California at Riverside
Regents of the University of California
200 University Office Building
Riverside California 92521

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFGA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-293

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Alan J. Akins/IFGA/(315) 330-1869/ Alan.Akins@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

The scope of this project was to explore the interplay between security and performance. We have succeeded in exploring a large cross section of ideas on work in a total of 12 different projects, which fall naturally into three distinct areas.
The first area deals with mechanisms for improving the performance of wireless ad-hoc networks. In this category, projects included improving TCP (Transmission Control Protocol) throughput, supporting group communication, routing, and on MAC (Media Access Control) layer enhancements for Mobile Ad Hoc Networks or MANETS.

The second general area concerns security. Our suite of projects in this domain cover the areas of key agreement, false report filtering, key establishment using pre-distribution, a public-key distribution mechanism using Secure DNS (Domain Name Service), and a study of issues in denial-of-service attacks in MANETs.

The final topic area involves the management of mobility. We have developed some very innovative approaches to handling complex spatio-temporal queries when objects move along road networks. This work represents a significant contribution to both wireless networking as well as to data management.

14. SUBJECT TERMS
Fault Tolerant Networks, Denial Of Service Attacks, Wireless Ad-Hoc Networks, Manets, Cyber Defense, Security, Mobility Management

15. NUMBER OF PAGES
62

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1. Improving TCP performance in Mobile Ad-Hoc Networks..................... 1

2. Group Communications for Mobile Ad Hoc Networks 9

3. Routing in Mobile Ad Hoc Networks... 17

4. MAC layer enhancements for Mobile Ad Hoc Networks...................... 23

5. Key Agreement for Dynamic Peer Groups.. 28

6. False Report Filtering in Mobile Sensor Networks................................ 30

7. Group-based Key Predistribution in Sensor Networks 33

8. Indexing Moving Objects: A cost model based approach...................... 36

9. Roads, Codes, & Spatiotemporal Queries 38

10. Efficient Data Dissemination Using Locale Covers............................... 40

11. Internet Key Service (IKS) 42

11.1 Our Contribution.. 43

11.1.1 RIKS: An IKS Prototype 45

12 Denial of Service Attacks at the MAC layer in MANETs..................... 46

List of Figures

Figure 1: The Experimental Set up. 2

Figure 2: Interactions between various layers and the proxy layer. 3

Figure 3: Improvement in TCP goodput versus maximum speed for one/two TCP connections 4

Figure 4: Goodput ratio vs distance between end nodes 7

Figure 5: Broadcast efficiency vs Rebroadcast size 10

Figure 6: Broadcast latency vs Rebroadcast size 10

Figure 7: Power adaptive broadcasting example 11

Figure 8: Energy consumption and percentage saving (n=4) 12

Figure 9: Logical links vs Physical links 13

Figure 10: Comparison of tree cost 13

Figure 11: Comparison of goodput vs group size 13

Figure 12: Comparison of packet delivery ratio under different group sizes in single source case 15

Figure 13: Comparison of total overhead with different number of sources 15

Figure 14: Probability of finding at least 3 node-disjoint paths, for various node densities 19

Figure 15: Probability of finding 3 node-disjoint paths, various robustness strategies 19

Figure 16: Routing table size vs network size in nodes. Logarithmic relationship. 21

Figure 17: Average path stretch factor vs network size. Constant with network size. 22

Figure 18: Communication and Storage Overheads 29

Figure 19: Probability of false report filtering in one hop 31

Figure 20: Fraction of false reports dropped vs. number of hops 32

Figure 21: Energy Consumption 32

Figure 22: Resilience and Communication Overhead Compared 35

Figure 23: A sample vTree 36

Figure 24: Join performance, varying dataset sizes. 37

Figure 25: Planar Road Network 38

Figure 26: Table Lookup vs Encoding 39
Figure 27: Time to compute Shortest Path 39
Figure 28: Example Locale Cover 40
Figure 29: 6-Nearest Neighbor Locale Cover for Chicago Dataset 40
Figure 30: Greedy vs Random Sampling 41
Figure 31: DOS attack in ad-hoc network. TCP throughput is negligible. 48
Figure 32: Use of FAIRMAC protects throughput during attack. 48
Figure 33: Tunable protection against aggressive UDP flows 49

List of Tables

Table 1: TGD and AFTD compared 29

Acknowledgements

This work in this report has been made possible by contributions from a number of people, whom we would like to acknowledge here.

First of all we list the students who formed the backbone of the effort, in no particular order: Gentian Jakllari, Jakob Eriksson, Lap Kong Law, Min Ge, Swastik Kopparty, Vasudev Shah, Vikram Gupta, Wenjie Luo, Xiaohu Chen, Zhenqiang Ye, Sandeep Gupta, Li Zhou, Swastik Kopparty, Jinfeng Ni, Saugat Majumdar, John Jones, Dan Berger, and Rui Jiang.

Second, we thank the various faculty colleagues here at UC – Riverside and elsewhere, whose feedback we found so helpful during out work.

Third, we acknowledge Dr. Douglas Maughan and Dr. Timothy Gibson, the Program Managers from DARPA, for recognizing the importance of the research area, and for their leadership.

Finally, we recognize our indebtedness to Mr. Alan Akins and Mr. Siamak Tabrizi of the Air Force Research Laboratory, the supervising agents for the work, for their active help, involvement, and encouragement throughout this work.

1. Improving TCP performance in Mobile Ad-Hoc Networks

It is likely that TCP will be used in ad hoc networks, since one might envision that ad hoc networks will ultimately have to be interfaced with the available wire-line and other wireless infrastructures. TCP provides transport layer reliability. Given that the high level objective of this work is to improve reliability and fault tolerance in ad hoc networks, it is imperative that we improve TCP performance. In this project we have undertaken a plurality of research tasks that are geared towards improving TCP in ad hoc networks.

Split TCP for Ad Hoc Networks: TCP has an inherent deficiency when used in ad hoc networks. It misinterprets packet losses due to link failures (which might be quite frequent due to mobility) as due to congestion. As a result there is a significant degradation in throughput. Furthermore, since link failures on long connections is more probable than link failures on connections of smaller length, shorter connections have an unfair advantage over longer connections when TCP is used. We proposed the use of *TCP proxies* to alleviate this problem. A long TCP connection is thus broken up (or split) into smaller segments and at the intersection of two segments we introduce what is known as a TCP proxy. The TCP proxy intercepts the TCP packet and sends an ACK on the behalf of the final destination to the source or to the previous proxy. It then takes over the responsibility of delivering the packet to the next proxy or to the destination as the case might be. This ***splitting*** of TCP session into smaller segments was seen not only to provide an increase in throughput but also to improve the fairness among the multiple TCP connections (each of varying length) that might exist in the ad hoc network. Furthermore, as an added benefit, if TCP were to be used in conjunction with the IEEE 802.11 MAC protocol, it was seen that the use of TCP proxies alleviates some of the problems that arise due to the interaction of TCP with the MAC protocol. Specifically, it is seen that when TCP is used in conjunction with the 802.11 MAC protocol, heavily loaded connections have an unfair advantage in access as compared to lightly loaded connections and connections that begin earlier might gain an unfair advantage as compared to connections that begin later. This is due to what is known as the capture effect whereby the TCP connection that is in advantage captures access rights in the local vicinity of the communication thereby other precluding other connections from accessing the bandwidth. Using proxies alleviates this problem by constraining the region that is captured to the region between proxies rather than between the source and the destination. Our results using the *ns 2.0* simulation package were presented at IEEE GLOBECOM 2002 [KKF02].

We also implemented Split TCP in a Linux-based experimental set up. We had six laptops and one of the laptops was configured to be the proxy. The node that acted as the proxy carried out the functionalities in terms of buffering packets and sending acknowledgements (mainly for the purposes of TCP transmission window control) to the sender. There was no need to maintain state since the proxy simply buffered all the packets for which it was a proxy and forwarded the packets along towards the next proxy and the destination without explicitly maintaining information with regards to each of the flows. A packet was simply discarded after an acknowledgement was received from the next proxy. We had an FTP (File Transfer Protocol) session from the source to the destination, and demonstrated that if a link failed (this was emulated by removing the PCMCIA card from the relay laptop that was not a proxy), the proxy continued to receive packets from the source and buffered the packets. If the link was restored, the packets only traversed the segment from the proxy to the destination. Proxies were implemented at a “pseudo-layer” between the IP layer and the TCP layer. Packets were nabbed just before they were forwarded at the IP layer, and buffered at the proxy layer. The proxy layer also handled congestion control functionalities of the transport layer protocol. We also have developed graphical user interfaces that enable us to visually appreciate what is going on. The experimental set up is as shown in Figure 1. We also show a block diagram in Figure 2 to show the interactions between the various layers.

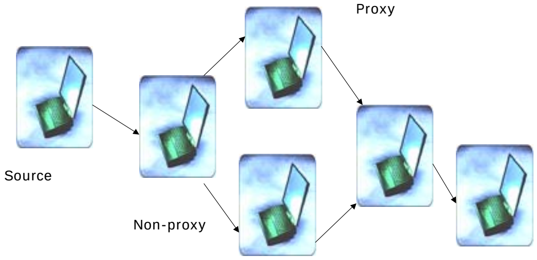


Figure 1: The Experimental Set up.

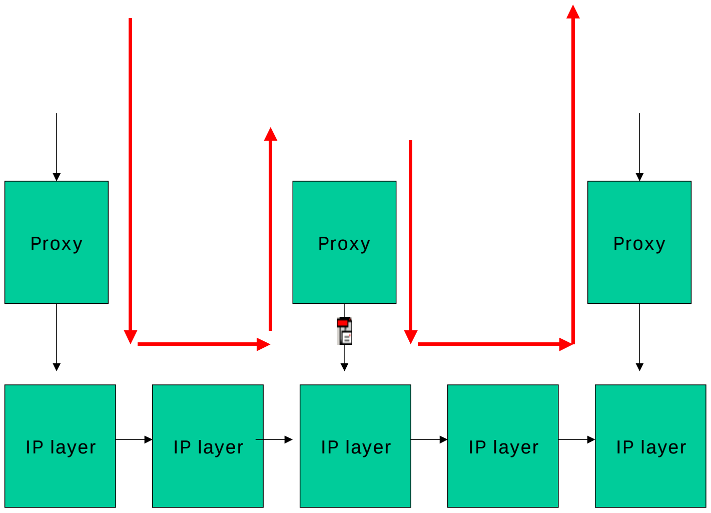


Figure 2: Interactions between various layers and the proxy layer.

A demonstration of our experimental set up was done at the DARPA PI meeting held in Newport, RI, in July 2003.

Signal Strength Based Link Management: TCP performs poorly in wireless ad hoc networks as demonstrated in [HV99, SX02]. The main reason for this poor performance is a high level of packet losses and a resulting high number of TCP retransmission timeouts. First, a node drops a packet if it cannot forward the packet to the next hop of the route as the next hop node has moved out of transmission range. A second reason for packet loss is congestion in the shared medium. In this case, a node cannot reach the next hop node because there are too many nodes trying to access the channel at the same time. This might even result in a node *capturing* the medium of access if the IEEE 802.11 MAC protocol is used [SX02]. While congestion can degrade the observed performance of TCP even in wire-line networks, mobility causes a degradation of performance of TCP in ad hoc networks even at very light loads. Our objective in this work was to mainly stem the degradation of TCP performance due to mobility.

Towards this goal, we proposed mechanisms to reduce the number of packet losses. These mechanisms are based on signal strength measurements at the physical layer. Based on these signal strength measurements, when a node *fails* to communicate with a neighbor, the MAC layer at the node guesstimates if the failure is due to congestion or due to the neighbor moving out of range. If the MAC layer predicts that the neighbor has just moved out of range, then it stimulates the physical layer to increase its transmission power and attempts to re-establish the link to the neighbor temporarily. It also prompts the routing layer to search for a new route. The signal strength measurements can also be used to predict possible link failures to a neighbor that is about to move out of range. Thus, if the measurements indicate that the signal strength is going down and the link is likely to break, a search for a new route can be proactively initiated before the link actually fails. While searching for the new route, the routing layer should take care to avoid either the temporary high power link or the weak link (as the case may be). We have made modifications to the ad hoc on-demand distance vector (AODV) routing protocol [PR99] such that it precludes the use of such links during the computation of a new route. In order to cope with failures that are not due to mobility, we have included a simple mechanism by which, the MAC layer, upon guesstimating that the neighbor is within range, *persists* in its attempt to reach that neighbor for a longer period of time. We re-iterate that our goals are mainly to cope with the effects of mobility on TCP. At high loads, it is more likely that congestion dominates packet losses. In the simulation experiments that we perform to evaluate our schemes, we therefore restrict ourselves to conditions of light load. In such scenarios we show that the performance of a TCP session can improve by as much as 45%. A sample result from our set of experiments is shown below in Figure 3. With our mechanisms, the improvements in TCP goodput are evident. More details can be found in [KYK05].

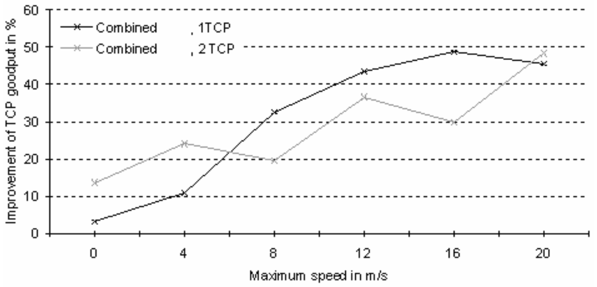


Figure 3: Improvement in TCP goodput versus maximum speed for one/two TCP connections

The use of signal strength and a count of the transmitted packets in the local neighborhood (nodes can overhear other packet transmissions) can provide an estimate of whether or not there is congestion in the local vicinity of a node. A node should only increase its transmission power if the network is lightly loaded. If there is congestion, temporary increases in power levels can actually increase the number of collisions and increase the congestion. This could degrade the performance further. We have also investigated the statistics of how the number of Request to Send packet transmission attempts (RTS), used with the IEEE 802.11 MAC protocol, vary for a given packet and our studies further confirm the validity of our approaches.

To the best of our knowledge, ours is the first work to use lower physical layer features such as signal strength and adaptive transmission power levels to improve TCP performance in ad hoc networks. The methods that we propose can be used with the User Datagram protocol (UDP) as well.

The efforts have lead to a conference and a journal publication. The paper was presented at the IFIP Personal and Wireless Communications Conference in Venice, Italy [KKT03] and an extended version will appear in the Ad hoc Networks Journal in March 2005 [KYK05].

Multipath Routing and Congestion-Aware Routing to Improve TCP Performance in Ad-hoc Networks: TCP sessions in ad hoc networks compete with each for bandwidth. The use of shortest path routing results in multiple TCP sessions being routed via a few congested areas or hotspots. Furthermore, congestion can cause nodes to *falsely* believe that links have failed when they have not. We have investigated the impact of both multi-path routing and congestion aware routing to understand if we can alleviate congestion and thereby improve TCP performance. Multi-path routing was seen to benefit mainly in terms of providing robustness to mobility, especially for long TCP connections but, did not provide any alleviation to congestion effects. The results of this work are summarized in [YKT04a].

We envisioned that spatially separating the TCP sessions such that they inflict much lower interference effect on each other could provide gains in performance. We constructed a centralized ideal approach that showed that the benefits due to spatial diversity do exist and they are limited only to long TCP connections (longer than 5 hops). We then designed a distributed congestion-aware routing scheme to exploit the potential benefits that we observed with the centralized ideal approach. In the distributed congestion-aware routing scheme, we define the number of TCP connections passing through a node and its one-hop neighbors as the congestion weight of this node. The congestion weight of a link is defined as the maximum weight of the two nodes that are connected by the link. The distributed congestion-aware routing scheme tries to discover a path with the least congestion weight.

Hello messages are introduced in order to exchange congestion weight among neighbors. We observe that even though the distributed congestion-aware routing scheme can help long TCP connections improving their goodput, it is at the expense of decreasing the goodput of short TCP connections. This observation is different from that with the centralized ideal scheme, in which the goodput of long TCP connections can be improved without hurting short TCP connections.

Even though the distributed scheme mimics the behavior of the centralized ideal approach, there were several differences between them. In order to understand the differences in behavior of the two schemes on TCP performance we considered several idealized distributed congestion-aware schemes:

1. Scheme I: In order to investigate the effects of HELLO messages on TCP performance, we consider an ideal scheme in which HELLO messages are magically exchanged, impromptu and without overhead.
2. Scheme II: In order to investigate the effects of stale congestion weight (of the neighbors) on TCP performance, we consider an ideal scheme in which the congestion weight of all nodes is magically updated once there are route changes (route breakage and route establishment) in the network. Note that HELLO messages are magically distributed without overhead as in the previous case.
3. Scheme III: Instead of propagating the congestion weight to one-hop neighbors, in this scheme we assume that the congestion weight of a node can be propagated to all its neighbors within the sensing range (as in the centralized scheme). Note that in this scheme HELLO messages are exchanged magically and the congestion weights of all nodes are updated at once, if there are route changes in the network.

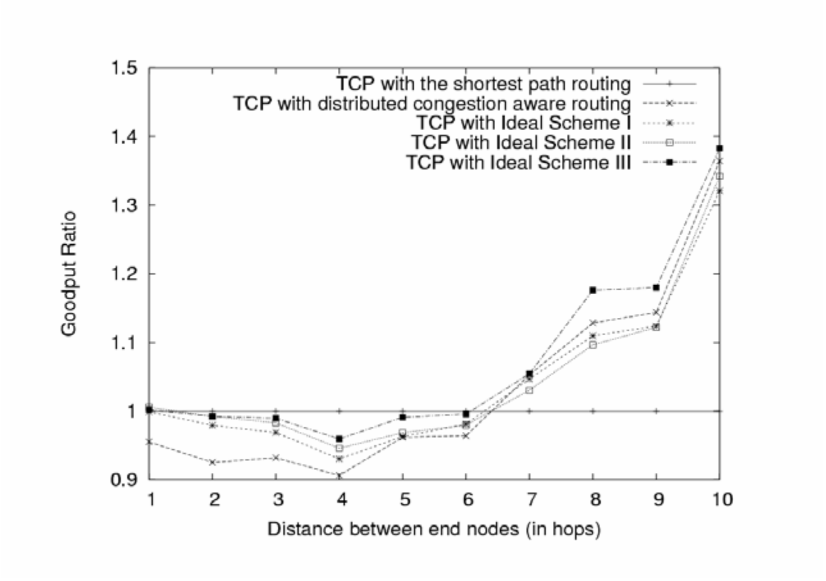


Figure 4: Goodput ratio vs distance between end nodes

Figure 4 shows the performance comparison of these idealized schemes with the realistic distributed congestion-aware routing scheme. The performance metric is the goodput ratio of TCP with the different schemes as compared with “TCP with the shortest path routing scheme”. We see that HELLO messages significantly hurt short TCP connections more than long TCP connections (ideal scheme I). After we eliminate the effects of HELLO messages and the stale congestion weight and, at the same time, extend congestion weight propagation range in Ideal Scheme III), the distributed version still performs worse than the centralized ideal scheme. The reason is that in the centralized scheme, a globally optimal path can be computed each time a route is needed. However, in the distributed scheme, due to lack of global information and information losses due to packet collisions, the discovered least congested path may not be the globally optimal path.

We conclude that even though we observe a certain level benefits due to spatial diversity for long TCP connections in the centralized ideal scheme, these benefits cannot be fully exploited in a distributed manner. The indispensable HELLO messages in order to exchange congestion information among nodes, lack of global information at each node, and information losses due to packet collisions prevent the distributed congestion-aware routing scheme from performing as well as the centralized ideal scheme (i.e., improving the goodput of long TCP connections without hurting short TCP connections).

Detailed discussions of the effects of multi-path routing and the effects of congestion aware routing on TCP performance can be found in [YKT03a] and [YKT03b], respectively.

2. Group Communications for Mobile Ad Hoc Networks

Group communications is an important communication paradigm in mobile ad hoc networks. Multicasting and broadcasting are obviously two of the most well-known group communications strategies that are employed in many applications. Due to certain constrains presented in MANETs such as the limited power supply, low available bandwidth and dynamic topology, the design of group communications protocol faces a considerable amount of challenges. Our objective of this work is to obtain a fundamental understanding towards group communications and develop certain lightweight and efficient multicast and broadcast protocols for mobile ad hoc networks.

Distance ADaptive (DAD) Broadcasting: In mobile ad hoc networks, it is often necessary to broadcast control information to all the constituent nodes in the network. Possible applications include searching for a destination node (as a part of routing) or service such as DNS look-up. Flooding, which is often deployed to achieve the above objective, is expensive in terms of overhead as it wastes valuable resources such as bandwidth and power. In dense network, flooding can be very inefficient and can cause significant contentions and collisions also known as the broadcast storm problem. A possible improvement to flooding is to choose probabilistically a subset of nodes to perform rebroadcast. The efficiency of broadcasting is primarily measured by the power it consumes for reaching the desired part of the network and the power consumption is then further related to the number of rebroadcasts that nodes have to be performed in order to complete the broadcast session. An idea to improve the broadcast efficiency is to select a portion of nodes that are further away from the previous broadcast node to perform rebroadcast. In this project, we proposed to use signal strength as an estimate of the relative node distance to optimized broadcasting operations.

In this project, we proposed an innovative way to improve broadcast efficiency by using the signal strength. We developed a Distance Adaptive (DAD) broadcast protocol to select a portion of nodes to perform rebroadcast depending on their relative distance from the previous broadcast node. We use signal strength as an estimation of the relative node distance to optimize broadcasting. Our goal is to maintain coverage, increase broadcast efficiency and reduce broadcasting latency.

DAD biases the selection of nodes to perform rebroadcast to those nodes that are far away from the previous broadcast node. Each node observes the signal strength of the receiving packets and those nodes that are further away from it have weaker signal strength. In this way, each node manages to select the outmost neighbors to perform re-broadcast. We proposed two variants of DAD depending on the way they select the rebroadcasting nodes:
DAD-NUM: We specify a certain number, k, of outmost nodes to perform rebroadcast.
DAD-PER: We specify a percentage, p, of the outmost nodes to perform the rebroadcast.
We performed a simulation on GLOMOSIM simulator to evaluate the performance of DAD under various mobility patterns and node densities.

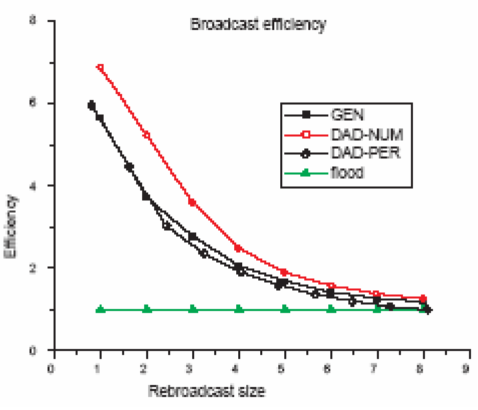


Figure 5: Broadcast efficiency vs Rebroadcast size

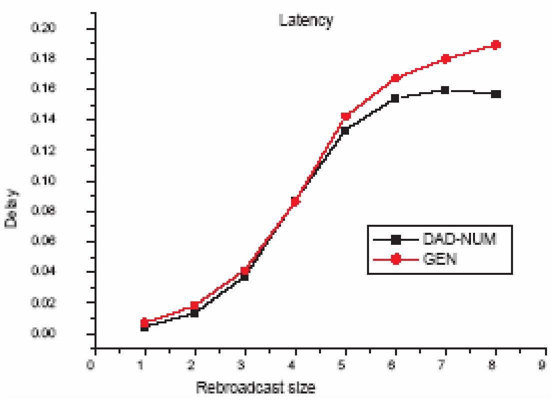


Figure 6: Broadcast latency vs Rebroadcast size

Our evaluation indicated that our approach can increase the efficiency of broadcasting significantly. With our approach, we can achieve the same coverage with approximately 20% less rebroadcasts compared to the probabilistic choice of rebroadcast nodes (see in Figure 5). In addition, the duration of the broadcast is also reduced by more than 21% (see in Figure 6).
This work was published at the IEEE MILCOM 2002 [Chen02].

Power Adaptive Broadcasting: Network wide broadcasting is an energy-intensive function. Despite of its expensiveness, it has been employed for numerous applications. Broadcasting is often being used for the dissemination of control messages or even used for the transmission of actual data. Reducing the overall energy consumption is extremely important in increasing the longevity of the network. However, most of the prior works on energy efficient broadcasting assumed that the originator of the broadcast has the global network information [Jeff00] [Liang02] (including both topology as well as the geographical distance between nodes in many cases). This can be prohibitive in terms of the overhead incurred. In this project, our objective is to propose a new method that performs local transmission power adaptation to reduce the overall energy consumed per broadcast.

The basic idea of our power-adaptive broadcasting protocol is to have nodes to reduce their transmission power range such that they reach only a sub-set of their neighbors and still maintain overall coverage. Each node determines if it is better, in terms of the energy consumed, to have a subset of its neighbors to relay broadcast packets to its other neighbors (outside the subset) instead of directly transmitting these packets to them in a single broadcast transmission. If it is determined to be better for a certain subset of its neighbors, the node will reduce its transmission power level to cover only this subset of neighbors. The nodes that are covered will then act as relays for the neighbors outside the subset. In the extreme case, a node might decide to reduce its transmission power to cover only its nearest neighbor.

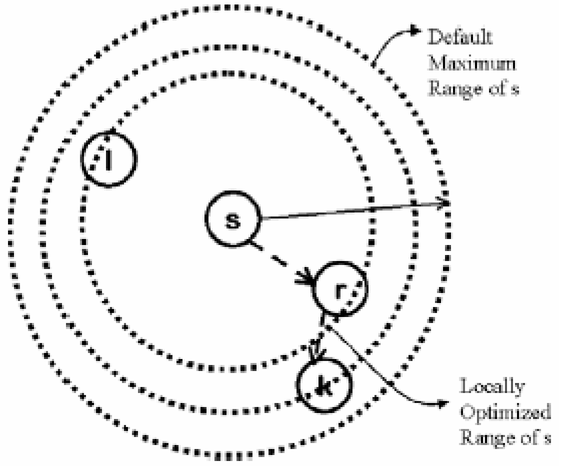


Figure 7: Power adaptive broadcasting example

As shown in Figure 7, node **s** is the originator of the broadcast. It figures out (from the two hop neighborhood information that is collected by means of Hello messages) that node **r** could potentially act as a relay of node **k**, i.e., node **k** can be covered when node **r** rebroadcasts. Thereby, node **s** reduces its transmission power range to exclude node **k** but still cover the next furthest node of its local range (in this case node **i**). Each node performs this local optimization when the node is required to perform a rebroadcast.

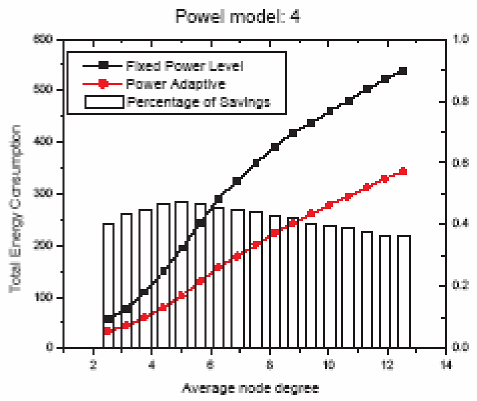


Figure 8: Energy consumption and percentage saving (n=4)

We performed an extensive simulation to compare our power-adaptive scheme with a non-power adaptive scheme and showed that our scheme achieves the same coverage as that of the scheme that does not perform power adaptations, but achieves a saving of about 40% in terms of energy consumption (see Figure 8).
This work was published at IEEE ICNP 2003 [Chen03].

Application Layer Multicast Algorithm (ALMA): In contrast to most of the previous researches on network-layer multicasting, we study the benefits of using application layer multicast in ad hoc networks. Using application layer approach has several potential advantages. First, it is easy to deploy and does not require any changes at the network layer. Second, the construction of the logical structure hides routing complications such as link failure instances, which are left to be taken care of at the routing layer. Third, intermediate nodes do not have to maintain per group state for each multicast group. Fourth, application layer multicast can exploit the capabilities of lower layer protocols in providing reliability, congestion control, flow control or security according to the needs of the application. However, the potential downside of using application layer multicasting is the redundancy of transmitting multiple copies of the multicast data packet over the same link due to the fact that non-multicast group members cannot make copies of the multicast packets. In this project, we intended to examine the trade-offs of using application layer multicasting in ad hoc networks. To do so, we developed an application layer protocol and studied its performance extensively.

The protocol that we developed is called Application Layer Multicast Algorithm (ALMA). ALMA creates a logical multicast tree between the multicast members.

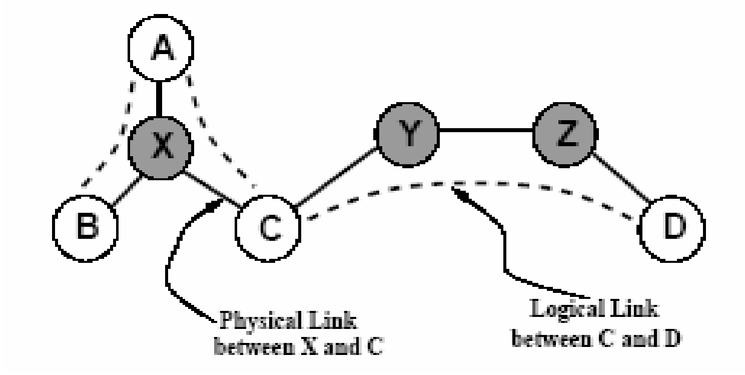


Figure 9: Logical links vs Physical links

From Figure 9, each dotted edge represents a logical link, which corresponds to a path at the network layer. As an example, there is a single logical link between C and D, but this logical link actually consists of three underlying physical links (C-Y, Y-Z and Z-D).

Due to the mobility of nodes, we provide an intelligent scheme to reconfigure the logical tree dynamically when the performance (quantified in terms of the observed round trip time from a child to its parent on the multicast tree) degrades beyond certain pre-specified thresholds. One of the key advantages of this approach is that the multicast structure need not be reconfigured at every instance of link failure. In lieu, the responsibility of the recovery of such failures is left to the lower layers.

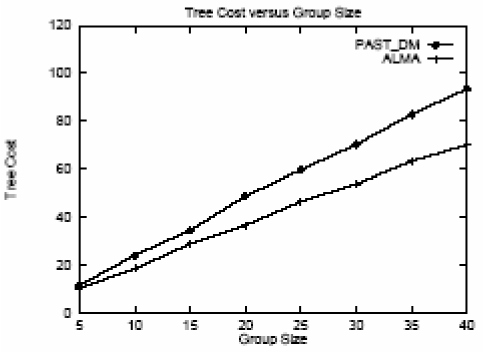


Figure 10: Comparison of tree cost

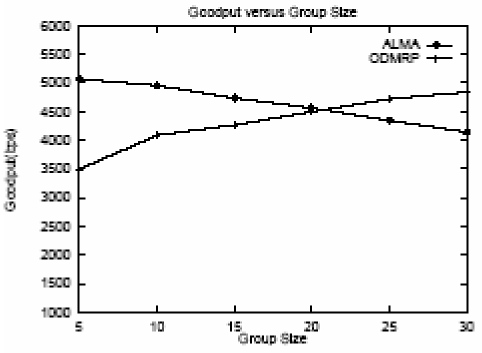
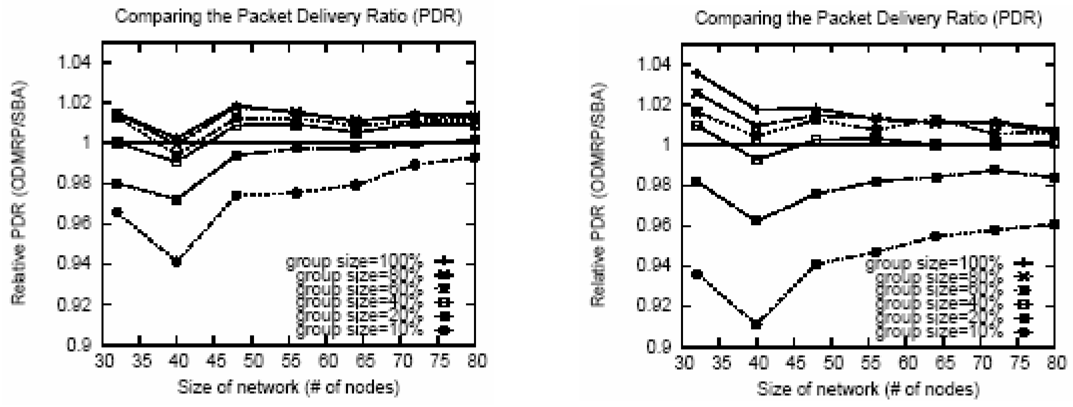


Figure 11: Comparison of goodput vs group size

We performed an extensive simulation to understand the benefits of ALMA. We showed that ALMA outperforms the best application layer multicast protocol, PAST-DM [Gui03], that is known to date for ad hoc networks. From Figure 10, we showed that ALMA outperforms PAST-DM in terms of the constructed tree cost. In order to ensure that

ALMA performs competitively with the network layer multicast protocols, we also perform extensive simulations to compare the performance of ALMA with that of the On-Demand Multicasting Routing Protocol or ODMRP [Lee00]. It has been shown that ODMRP is one of the (if not the best) network layer multicast routing protocols proposed for ad hoc networks. We observed (from Figure 11) that ALMA outperforms ODMRP for small group sizes (when the number of multicast group members is small). However, its performance degrades when the size of the multicast group increases. In general, ODMRP outperforms ALMA when the group size is greater than 46%. For extremely large group sizes ODMRP clearly outperforms ALMA but for these extremely large group sizes, performing multicasting is in fact close to performing broadcasting. This work was published at Med-Hoc-Net 2004 [Ge04].

Understanding the trade-offs between broadcasting and multicasting: Typical multicast protocols for mobile ad hoc networks (MANETs) typically require the use of flooding control messages to create and maintain a multicast structure for data distribution. However, it could potentially be extremely heavy-weight since a periodical invocation of control messages with high frequency is required to maintain an up-to-date multicast structure in the scenarios of high mobility wherein the information tends to stale fairly quickly. One might expect in such scenarios, broadcasting might be a more attractive option since it is more robust to mobility and does not require any structure be constructed beforehand. In addition, because of the use of a shared wireless media, it is even more beneficial to use broadcasting in the scenarios of high group density as a single broadcast of a data is received by all nodes within the transmission vicinity. However, the benefit of broadcasting is limited and will again become inefficient in the scenarios wherein the number of group members is relatively few. Obviously, there are some trade-offs between the use of broadcasting and multicasting in MANETs but theses trade-offs are not well studied and quantified. Therefore, the problem is to quantitatively evaluate the trade-offs of them when they are deployed in different network scenarios.



Speed = 10m/s

Speedy = 20m/s

Figure 12: Comparison of packet delivery ratio under different group sizes in single source case

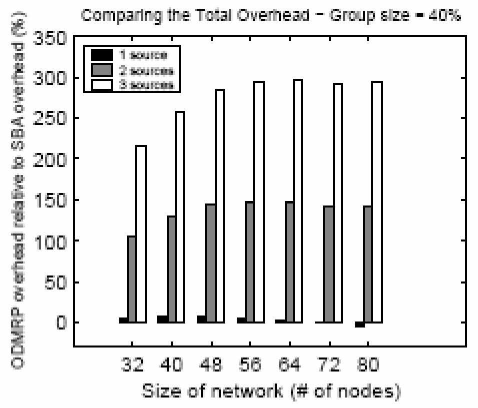


Figure 13: Comparison of total overhead with different number of sources

To evaluate the trade-offs between broadcasting and multicasting, we considered to compare an intelligent broadcast scheme called Simple Broadcast Algorithm (SBA) [Peng00] and a well-known multicast scheme called On-Demand Multicast Routing Protocol (ODMRP) [Lee00] under different network scenarios. Our evaluation is conducted by simulations using the *ns-2* simulator. We compare them based on the packet delivery ratio and overheads. Our evaluation indicates that there is no clear winner between broadcasting and multicasting and the choice is purely based on the scenario in question. Broadcasting is more preferable to use in the scenarios where the group size is greater than 40% or when the mobility is high (see Figure 12). Multicasting is more preferable to use in the scenarios where the group size is smaller than 40% and when the mobility is low. Moreover, broadcasting seems to be more preferable when the number of sources is large since the benefit of having no structure constructed in broadcasting outweighed the benefit of reducing unnecessary data forwarding in multicasting (see Figure 13). The fundamental reason is the high overhead incurred for multicasting to construct different multicast structures for different sources.

This work was published at IEEE MILCOM 2004 [Law04].

3. Routing in Mobile Ad Hoc Networks

Routing protocols play an integral part in any ad hoc network that requires connectivity over more than a single radio range. Current routing protocols lack robustness to node failure, and are not sufficiently scalable to support networks over a few hundred nodes, in most scenarios. In our routing work, we provide a framework for reliable routing in MANETs, as well as a novel routing protocol in which control overhead scales logarithmically with network size, enabling the creation of very large ad hoc networks.

A Framework for Reliable Routing in MANETs. Mobile ad hoc networks consist of nodes that are often vulnerable to failure. As such, it is important to provide redundancy in terms of providing multiple node-disjoint paths from a source to a destination. We first proposed a modified version of the popular AODV protocol that allows us to discover multiple node-disjoint paths from a source to a destination. We found that very few of such paths can be found. Furthermore, as distances between sources and destinations increase, bottlenecks inevitably occur and thus, the possibility of finding multiple paths is considerably reduced. We concluded that it is necessary to place what we call reliable nodes (in terms of both being robust to failure and being secure) in the network for efficient operations. We proposed a deployment strategy that determines the positions and the trajectories of these reliable nodes such that we can achieve a framework for reliably routing information. We defined a notion of a reliable path which is made up of multiple segments, each of which either entirely consists of reliable nodes, or contains a preset number of multiple paths between the end points of the segment. We showed that the probability of establishing a reliable path between a random source and destination pair increases considerably even with a low percentage of reliable nodes when we control their positions and trajectories in accordance with our algorithm.

Mobile ad hoc networks find application in many fields such as military deployments, disaster rescue missions, and electronic classrooms. We primarily look at reliability in terms of providing robustness to node failures in ad hoc networks. Node failures may be intermittent, i.e., for short periods or for long periods of time, and due to various reasons. First, since these networks are likely to be deployed in wireless environments, the communications between the ad hoc nodes will have to be via a harsh fading channel. Thus, communications between nodes would typically endure periods of intermittent failure and as a consequence, packet losses. It is possible that certain nodes might completely lose connectivity for temporary periods due to the fading conditions. One way of overcoming this would be to use sophisticated antenna systems or modulation methods. However, many of the ad hoc nodes, if not most of them, would be constrained by size, processing and power limitations and thus, may not possess such capabilities. Second, many of the ad hoc nodes are power constrained. Due to battery drain, it is possible that some of these nodes might not be able to function. Such an effect may result in a long term failure if a node's battery is completely drained or if it is possible to re-charge the node's battery, the node might not function for intermittent short periods. Third, nodes in an ad hoc network are vulnerable to compromise. Compromises are especially likely for unattended sensor nodes or handhelds carried by pedestrians. A simple form of denial of service is to simply cause node failures, either intermittent or long term.

Multipath routing is one way of improving the reliability of the transmitted information. While multipath routing may be used for various other reasons such as load-balancing, congestion avoidance, lower frequency of route inquiries and to achieve a lower overall routing overhead [Maxemchuk75][Pearlman00][Nasipuri01][Marina01][Wu01], our objective is to primarily design a multipath routing framework for providing enhanced robustness to node failures. If one could provide multiple paths from a source to a destination, one could envision the transmission of redundant information on the various paths (by the use of known techniques such as diversity encoding [Ayanoglu93]) that would help the receiver in reconstructing the transmitted information even if a few of the paths were to fail. By multiple paths, we imply multiple node-disjoint routes from a source node to a destination node. Our first goal towards this was to design a routing protocol that would allow us to find multiple node-disjoint paths from a given source to a destination. Towards this, we made modifications to the Ad Hoc Distance Vector Routing Protocol (AODV) which is one of the most popular ad hoc routing protocols to facilitate the discovery, and the use of multiple node-disjoint paths.

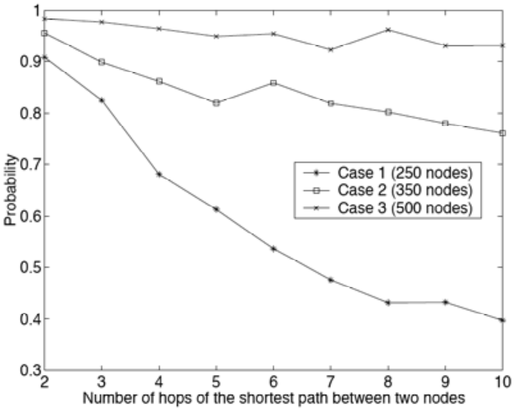


Figure 14: Probability of finding at least 3 node-disjoint paths, for various node densities.

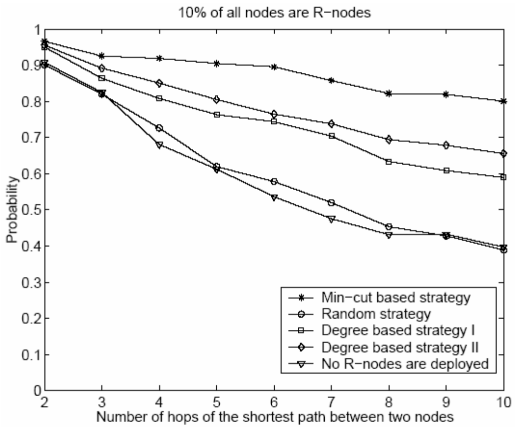


Figure 15: Probability of finding 3 node-disjoint paths, various robustness strategies

We found that the number of node-disjoint paths from a source to a destination is dependent on the node density in the ad hoc network (as might be expected). Furthermore, we found that as the distance between a source and its destination is increased, one could find no more than a very limited number of paths between them, even at moderate node densities (average node degree is 6.7). This observation leads us to believe that, one would require at least a few of the ad hoc nodes to be more reliable. One could envision that these nodes would be placed in moving vehicles and could be less constrained in terms of size, processing and power. They would be physically more secure and robust to compromises. These nodes (typically much fewer in number in comparison with the normal ad hoc nodes) could then, be allowed to participate in routing along multiple routes between the same source-destination pair. For the ease of notation let us call these nodes R-nodes. The revised objective is then to construct a sequence of reliable segments between the source and the destination. Nodes that join two segments have to be R-nodes. A segment is deemed reliable if it consists of either a preset number of paths between the two R-nodes that it connects or if it is made up of R-nodes entirely. A concatenation of reliable segments is called a reliable path.

The next question that arises is: where should these R-nodes be placed so that the probability of finding a reliable path between an arbitrary source and destination is acceptable? Initially, we placed these R-nodes at random locations within the area of interest. However, we found that this does not help in achieving an acceptable probability of finding a reliable path between a source and a destination. Thus, we need a more

intelligent way of placing these R-nodes. Furthermore, as the nodes in the ad hoc network are mobile, the R-nodes would have to adaptively move so as to maintain these advantageous positions with respect to the other nodes. We proposed a methodology to control the trajectory of an R-node based on information exchanged within a local vicinity of the R-node. We found by simulations that placing each R-node at positions defined by our algorithm (which is in fact, a version of the randomized min-cut algorithm [Motwani95]) is a very effective deployment strategy in terms of achieving a high probability that a reliable path is found between any arbitrary source and destination.
More details on this work can be found in [Ye03].

Dynamic Addressing for Routing Scalability in MANETs. Scalability is a critical requirement in the use and deployment of ad hoc networks, if we want this technology to reach its full potential. Ad hoc networking technology is receiving a lot of interest but it has yet to mature. This is similar to the early stages of the Internet, where very few could predict its explosive growth. A difference is that in the Internet, scalability was, from the very beginning, a design constraint. Ad hoc networks research seems to have downplayed the importance of scalability. In fact, current ad hoc architectures do not scale well beyond a few hundred nodes.

How can we make ad hoc networks scale to thousands, or even millions of nodes? We found this question fundamental if we want ad hoc technology to be successful in the consumer marketplace. Already, non-military technology and applications seem to point towards future networks with: a) ad hoc pockets of connectivity, b) consumer-owned networks, and c) sensor-net technologies. All of these applications will place increased scalability demands on ad hoc routing protocols. Most current research in ad hoc networks focus more on performance and power-consumption related issues in relatively small networks, and less on scalability. The current routing protocols and architectures work well only up to a few hundred nodes. We believe the main reason behind the lack of scalability is that these protocols rely on flat and static addressing. With scalability as a partial goal, some efforts have been made in the direction of hierarchical routing and clustering [Ramanathan98] [Pei99] [Pei00]. These approaches do hold promise, but they do not seem to be actively pursued. It appears to us as if these protocols would work well in scenarios with group mobility [Hong99], which is also a common assumption among cluster based routing protocols.

Is dynamic addressing a feasible way of achieving scalable ad hoc routing? This is the question that we address in this work. Dynamic addressing simplifies routing but introduces two new problems: address allocation, and address lookup. In this project, we focused on the address allocation part; earlier work describes a general idea of how address lookup can be efficiently handled [Eriksson03]. As a guideline, we identified a set of properties that a scalable and efficient solution must have:

- Localization of overhead: a local change should affect only the immediate neighborhood, thus limiting the overall overhead incurred due to the change.

- Lightweight, decentralized protocols: we would like to avoid concentrating responsibility at any individual node, and we want to keep the necessary state to be maintained at each node as small as possible.

- Zero-configuration: we want to completely remove the need for manual configuration beyond what can be done at the time of manufacture.

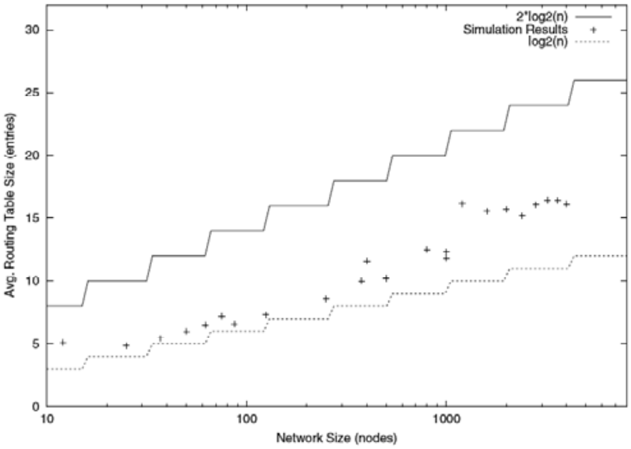


Figure 16: Routing table size vs network size in nodes. Logarithmic relationship.

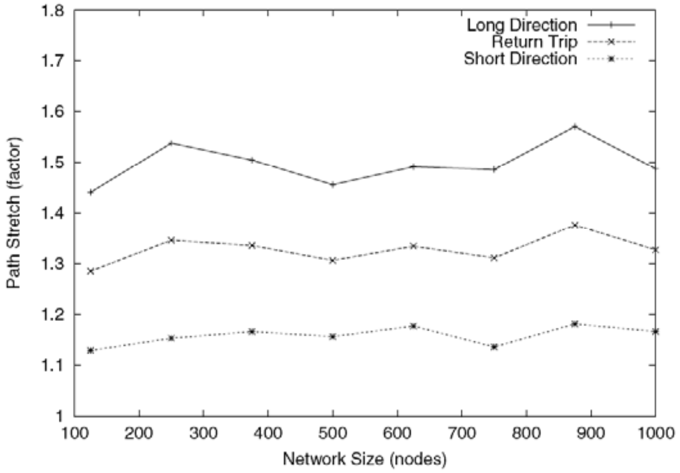


Figure 17: Average path stretch factor vs network size. Constant with network size.

In our work, we evaluated dynamic addressing and show that it is a promising first step toward achieving scalability in the order of millions of nodes in ad hoc routing. First, we developed a dynamic addressing scheme, which has the necessary properties mentioned above. Our scheme separates node identity from node address, and uses the address to indicate the node's current location in the network. Second, we studied the performance of a new routing protocol, based on dynamic addressing, through analysis and simulations.

In more detail, our work leads to the following results.

- Our address allocation scheme uses the address space efficiently on topologies of randomly and uniformly distributed nodes, empirically resulting in average routing table sizes of less than 2*log2(n) where n is the number of nodes in the network.
- Our dynamic addressing based routing scheme provides good network performance. In fact, our results indicate that we would reliably outperform other routing protocols based on static addresses, in large and actively used networks.

Our work in perspective. We provided a new approach to routing in ad hoc networks, and compared it to other current routing architectures. However, the goal was to show the potential of this approach and not to provide an optimized protocol. We believe that the address equals identity assumption used in current ad hoc routing protocols is most likely inherited from the wireline world, which is much more static and is explicitly managed by specialist system administrators.

4. MAC layer enhancements for Mobile Ad Hoc Networks

The medium access control layer is extremely important towards achieving high levels of performance in ad hoc networks. The reliability and efficiency at the MAC layer greatly impacts the performance at the higher layers. We have conducted research at the MAC layer on three distinct topics:

TCP Friendly MAC layer design
MAC in power heterogeneous ad hoc networks
Polling based MAC for use with array antennas.

The first topic is geared towards alleviating some of the problems that TCP faces when deployed in mobile ad hoc networks; in particular, we have focused on self-contention, the phenomenon in which TCP packets of the same flow contend with one another. In the second topic, we recognize that power heterogeneity is a reality in tactical deployments since nodes are likely to operate with differing transmission power levels (sensors versus bigger nodes housed in vehicles). Current MAC protocols are not equipped to work efficiently in such scenarios. We have proposed modifications to the IEEE 802.11 MAC protocol to improve its performance in the face of heterogeneity. Finally, we design a MAC protocol for use with directional antennae given that these are likely to be deployed in ad hoc networks.

TCP Friendly MAC Layer Design: Multi-hop communications as in ad hoc networks introduces the unique problem of *self-contention*, caused by contention between packets of the same transport connection at different nodes. The problem of self-contention arises due to uncoordinated access to the shared media by different nodes. Specifically, one of the reasons for degraded TCP performance is the contention caused by TCP-DATA packets on the TCP-ACK packets and vice versa. Moreover, multiple TCP-DATA or ACK packets, belonging to the same flow, and flowing in the same direction would contend for bandwidth among themselves. The IEEE 802.11 MAC protocol, primarily designed for wireless LANs fails to cope with these multi-hop effects. As a result, some of the nodes back-off frequently and thereby can cause TCP connections to throttle.

We have proposed two MAC layer extensions that are designed with TCP in mind. We once again emphasize that one of the main reasons for the deterioration of TCP performance is because of what we call self-contention: contention between the TCP packets of the same flow. A TCP-DATA packet contends for bandwidth with immediately preceding and succeeding DATA packets from the same flow and with the TCP-ACK packets of the flow that return from the destination to the source. This self-contention, with the legacy IEEE 802.11 MAC protocol can cause nodes to frequently back-off and thereby can throttle the TCP connection. We propose two specific techniques to deal with self-contention: the ***Quick-Exchange Technique*** and the ***Fast-Forward Technique***. Both of the techniques are based on the provision of simple yet effective TCP friendly extensions to the IEEE 802.11 MAC protocol. We see from our simulation studies that these techniques offer promise in terms of improving TCP performance in some regular topologies. However, some of the idiosyncrasies of TCP do not make Fast Forward work well in randomly generated networks.   The detailed results from our studies are found in [BYS04].

The *quick-exchange (QE) extension* allows for the exchange of two packets flowing in opposite directions, between adjacent nodes, in a single dialog (Request To Send or RTS and Clear To Send or CTS exchange). The RTS-CTS-DATA-ACK dialog (traditionally used with the IEEE 802.11 scheme) is extended by an additional data packet transmission (DATA2) from the RTS-receiver. The acknowledgment (ACK1) for the first data packet (DATA1) is *piggybacked* onto DATA2. Finally, if DATA2 is received correctly, the RTS-sender sends a corresponding acknowledgment (ACK2) back to the DATA2 sender. As mentioned earlier, the objective of using QE would be to piggyback returning TCP-ACK packets (as DATA2 packets) during the MAC layer transfer of TCP-DATA. Similarly, TCP-DATA could be piggybacked onto an ACK transfer at the MAC layer.

The fast-forward modification (FF) introduces a new packet type (ACK-RTS), which serves both as a MAC-layer acknowledgment as well as an RTS message.   Upon receipt of a data packet header, the receiving node determines, via a request to the routing layer, the next hop for the incoming packet.   If the next hop can be determined without the need to send route discovery traffic, the receiving node sends an ACK-RTS, simultaneously informing the sender of successful receipt and announcing the intent to forward this packet. The node targeted by the ACK-RTS packet responds per the normal 802.11 RTS/CTS mechanism, and all nodes, which receive the ACK-RTS update their Network Allocation Vectors with the encoded duration. In the event that no CTS is received in response to the ACK-RTS, or if transmission fails, the data packet is delivered to the higher layers and normal behavior is resumed.

The simulations that we performed show that QE provides great promise and can help increase the TCP goodput by 15 % or more in string, grid and random topologies. While the FF scheme does provide improvements with the string topology, it fails to provide improvements with the random topology. This is due to complex interactions between the FF mechanism and the round trip time estimation mechanism of TCP.

Our research was presented first as a poster in MOBICOM 2003 [YBS03] and then as a full paper in the IEEE conference on Mobile Ad hoc and Sensor Systems (MASS 2004) in Fort Lauderdale [DBS04]. A more detailed discussion appears in these papers.

Media Access Control in Power Heterogeneous Ad Hoc Networks: All Medium Access Control (MAC) layer protocols for wireless ad hoc networks typically assume that the network is homogeneous with respect to the transmit power capability of individual nodes in the network. In other words there exists symmetrical links in the network. The rapid spread of diverse “wireless network facilitated” devices endangers the assumption of homogeneous power capability. An ad hoc network may comprise low power transducers, Personal Digital Assistants (PDAs), handheld computers and larger file servers. These devices will have different transmitting power capabilities. Some of them may be “tethered” to a power supply at all times and others may be dependent on battery power for long durations of time. In any case, it will be critical to ensure that the MAC protocol in use does not unjustifiably favor devices that can transmit at higher power levels. We describe some of the issues associated with using the IEEE 802.11 MAC protocol in a network in which different nodes may transmit at different power levels.

The performance of the IEEE 802.11 MAC protocol has been shown to degrade considerably in an ad hoc network with nodes that transmit at heterogeneous power levels. The main cause of this degradation is the potential inability of high power nodes to hear the RTS/CTS exchanges between nodes when at least one node involved in communication is a low power node. The propagation of the CTS message beyond the one-hop neighborhood of two communicating low power nodes was considered in our prior work in an attempt to alleviate this effect. However, this resulted in excessive overhead and further degraded the performance at the MAC layer.

In our work we considered two techniques to reduce the overhead incurred due to the aforementioned propagation of the CTS message: (a) the use of an intelligent broadcast scheme and (b) the reservation of bandwidth for the sequential transmission of multiple data packets with a single RTS/CTS exchange (and propagation as needed). These techniques required changes only at the MAC layer. We found, by means of extensive simulations, that the techniques provide a significant improvement over the original 802.11 MAC protocol in the considered *power heterogeneous* ad hoc network. The overall throughput improves by as much as 12 % and the throughput of the low power nodes improves by up to 14 % as compared to the IEEE 802.11 MAC protocol. Furthermore, the schemes find applicability even in homogeneous networks as they reduce the number of *false link failures* that arise when the IEEE 802.11 MAC protocol is used, by about 20 %. We concluded that the schemes together offer a simple yet effective and viable means of performing medium access control in power heterogeneous ad hoc networks. Our work was presented in IEEE ICC 2004 and further details may be found in [SKP04].

Polling based Media Access Control for use with Array Antennas: Directional Antennae can help abate interference effects by either focusing the transmission energy in a particular direction or by tuning the antenna to receive the energy from a particular direction, or by doing both of the above. The use of directional antennae is especially attractive for military networks.

Previous MAC schemes designed for use with directional antennae rely on omni-directional transmissions of control messages by nodes that try to reconnect with neighbors that move out of their angular range. Furthermore, most of the previously proposed schemes restrict themselves to either only directional transmissions or directional receptions. The inability of exclusively using directional antennae for both the transmission and reception of all MAC layer frames (control or data) results in two major problems: (a) the spatial re-use benefits are reduced due to the invocation of omni-directional communications and (b) the use of omni-directional receptions for certain packets and directional receptions for others leads to an inherent *asymmetry in range*. This phenomenon can exacerbate the hidden terminal problem [RR02] and leads to a significant penalty in throughput.

A challenge associated with the exclusive deployment of directional antennae for all communications in mobile networks is that, due to the angular reduction in range in comparison to the omni-directional case, it is important for a node to *poll* each of its neighbors periodically to ensure that the neighbor's motion is tracked. The MAC protocols proposed thus far either completely ignore mobility or use omni-directional transmissions or receptions (thus inflicting the asymmetry in range problem) of HELLO messages to identify neighbors.

In the preliminary work that we could do under this project, we proposed a new MAC protocol for mobile ad hoc networks that addresses the issues mentioned above in an integrated way. We call our protocol PMAC for *Polling-based MAC protocol*. PMAC exclusively uses directional antennae for the transmission and reception of all frames, i.e., we obviate omni-directional transmissions and receptions. Furthermore, the protocol facilitates the discovery of new neighbors by a node, and using polling, the maintenance of links to the discovered neighbors until they are outside the possible radial range of the node. Polling is also used to schedule the transmissions and receptions of information. This would ensure that the transmitter and the receiver nodes point their antenna elements towards each other at the time that they are scheduled to communicate. The preliminary work has been submitted as a paper to IEEE WOWMOM 2005, for consideration [JLK05].

5. Key Agreement for Dynamic Peer Groups

Problem Description

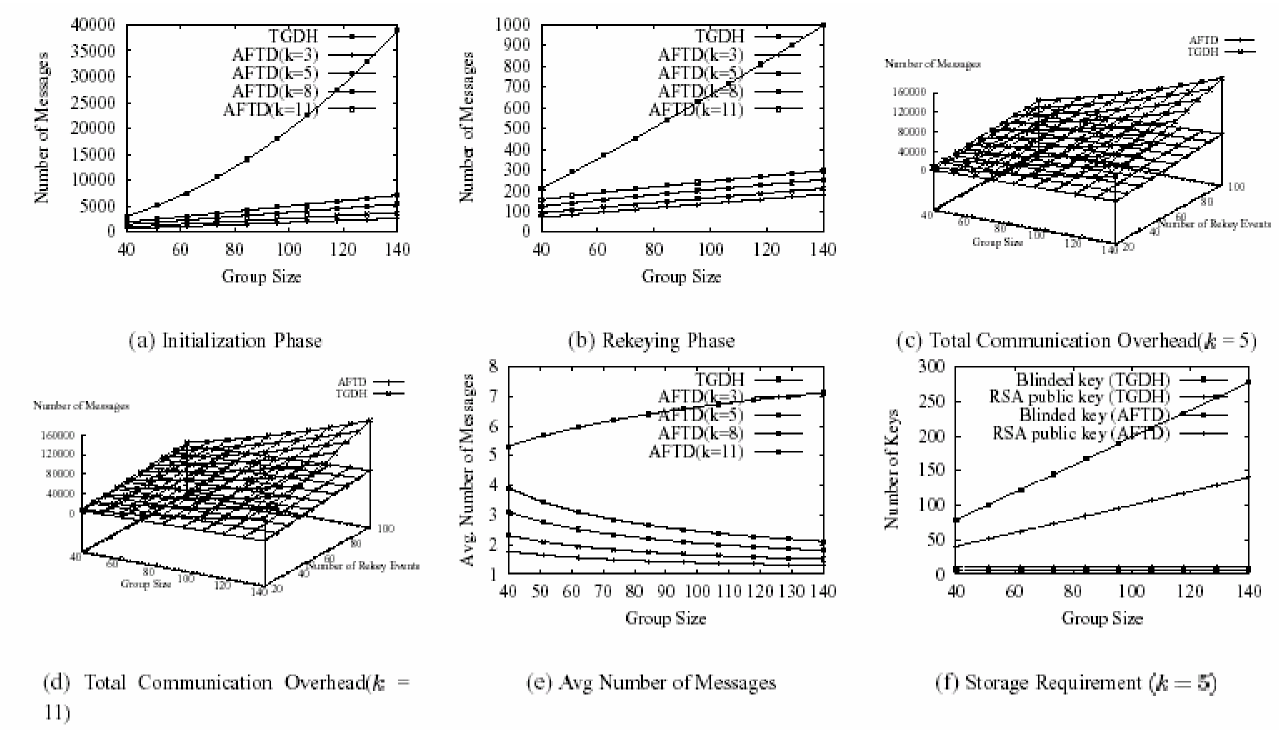
As a result of the increased popularity of group-oriented applications, there is a growing demand for security services to achieve secure group communication. A common method is to encrypt messages with a group key, so that entities outside the group cannot decode them.

The group key is updated on every membership change for forward secrecy and backward secrecy. This method is called *group rekeying*. To reduce the number of rekeying operations, Wong et al. proposed a logical data structure called a *key tree* [Wong98] that reduces the rekeying overhead from *O(n)* to *O(log n)*, where *n* is the group size. Based on this idea, Kim et al. proposed a tree-based key agreement protocol, TGDH [Kim00], which is a combination of key tree and Diffie-Hellman key exchange [Diffie76] to generate and maintain the group key.

Unfortunately, TGDH suffers from two drawbacks. It remains prone to impersonation attacks, and uses more messages than necessary.

Our Contribution

In this work, we propose a novel Authenticated, Fault-tolerant Tree-based Diffie-Hellman key agreement protocol, AFTD, based on two key ideas. First, it is gross overkill to broadcast updated public keys to all group members for recomputing the group key when a node n_i joins or leaves. It suffices to send each update to a much smaller subset of nodes in the tree, called its *trust set* $TS(n_i)$. Second, we achieve robust key authentication by distributing the function of trusted authority among the nodes in $TS(n_i)$, using a threshold cryptographic scheme. Any *k* members of a node's trust set can serve as its public key certificate authority. Our performance analysis shows this scheme can reduce the communication overhead from $O(n^2)$ to *O(nlogn)* for initialization, and from *O(nlogn)* to *O(n)* for rekeying. It also reduces the storage requirement for blinded keys from *O(n)* to *O(logn)*. (See Figure 18 and Table 1) This feature is particularly useful when a broadcast channel is unavailable.



(a) Initialization Phase

(b) Rekeying Phase

(c) Total Communication Overhead($k = 5$)

(d) Total Communication Overhead(k = 11)

(e) Avg Number of Messages

(f) Storage Requirement ($k = 5$)

Figure 18: Communication and Storage Overheads

	Communication Overhead		Storage Requirement	
	Initialization Phase	Rekeying Phase	Blinded Key	RSA Public Key
TGDH	*$O(n^2)$*	*O(nlogn)*	*O(n)*	*n*
AFTD	*O(nlogn)*	*O(n)*	*O(logn)*	*2k*

Table 1: TGD and AFTD compared

6. False Report Filtering in Mobile Sensor Networks

Problem Description

Sensor networks commonly deploy many small and inexpensive sensing devices over a large field to sense events or obtain readings of parameters. Sensor networks may be deployed, for example, to detect forest fires, unusual traffic patterns, or environmental changes. Typically, a sensor detecting an event will send a report to a special node called a *sink*, which collects and processes such reports. When the events of interest are critical events, reports must be delivered securely to the sink. In practice, however, physical security is difficult since sensors are prone to capture, and strong cryptographic methods may require more resources than are typically available. Once a node has been compromised, the adversary has access to all data and cryptographic keys stored at the node. At this point, the adversary may have enough key information to impersonate the compromised node and mount a variety of attacks.

We present a secure mechanism to prevent false report attacks [Ye04], mounted by an adversary who gains control over a node's resources and generate false reports appearing to originate from uncompromised nodes. Such false reports will waste scarce resources such as energy and bandwidth, but more importantly, may also lead the sink to make wrong decisions, having serious high-level consequences. For example, a false report of an emergency situation would send first-responders to the wrong location, depriving legitimate requests of urgently needed resources.

Consequently, false reports should be filtered as early as possible as they are transmitted to the sink.

A security scheme for false report filtering in sensor networks should be efficient, scalable, and localize faults. However, most existing approaches [Ye04, Zhu04, Yang04] cannot localize the impact of node compromises.

Our Contribution

In this work, we present two fault-localized schemes for false report filtering, that are efficient and scalable. Our schemes utilize one-way hash chain for each detecting sensor, using which en-route nodes are able to verify the authenticity of received reports, based on key commitments by detecting sensors. They differentiate the roles of detecting nodes and en-route nodes, while limiting the impact of node compromise to its locale. In addition, as a consequence of fault localization, our schemes enable localized protection when some important areas require special protection. Our second scheme is an enhancement of the first one, and is suitable for mobile sensor networks, using a random commitment predistribution scheme.

Our security analysis shows that more than 98% false reports are dropped within 2 hops when false reports are generated by a compromised cluster head. In the worst case, more than 90% false reports can be filtered within 8 hops (Figure 19). Further, although our scheme introduces extra fields in report packet, they result in very considerable energy savings by filtering most false reports very quickly (Figure 20).

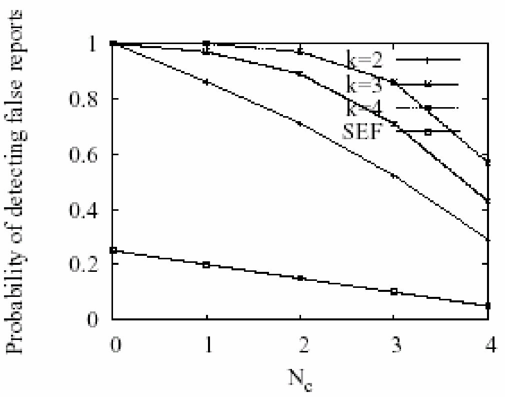


Figure 19: Probability of false report filtering in one hop

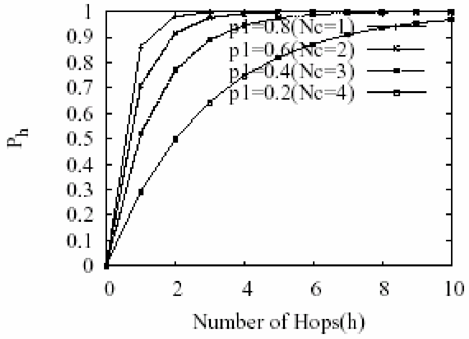


Figure 20: Fraction of false reports dropped vs. number of hops

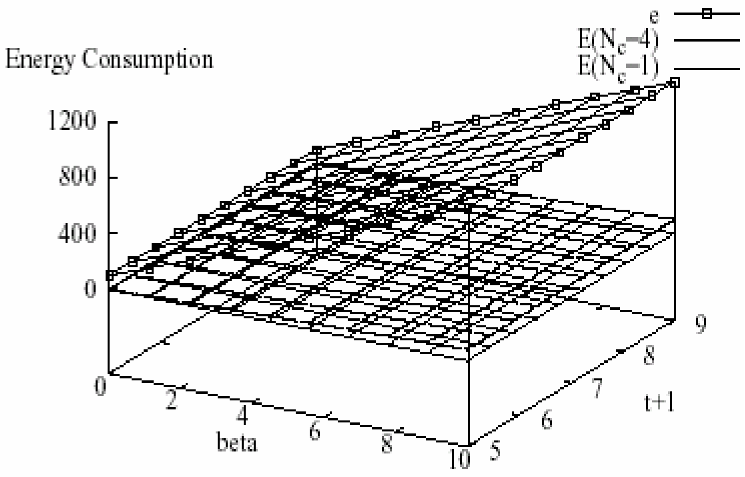


Figure 21: Energy Consumption

7. Group-based Key Predistribution in Sensor Networks

Problem Description

Sensors typically need to communicate with their neighboring sensors, aggregating sensing data into a more compact report and then transmitting it to the base station over multiple hops. Messages sent between neighboring sensors may contain sensitive data or commands from the base station, thus it is crucial to secure these communications. Unfortunately, resource limitations at sensor nodes rule out the use of expensive public key cryptosystems such as RSA [Rivest78] or Diffie-Hellman key agreement [Diffie76] for this purpose.

Key predistribution schemes are typically considered as an efficient way to establish pairwise keys between neighboring sensors. This task, however, is complicated by the ad-hoc and on-demand nature of sensor deployments. Since a sensor's neighbors are only known after deployment, it is not possible to preload these shared keys in any simple way.

Recently, some random key predistribution schemes [Eschenaer02, Chan03, Du03, Liu03, Du04] have been proposed for large-scale sensor networks. However, these random key predistribution schemes suffer from two major problems, which render them inappropriate in some applications. First, these schemes require the deployment density to be high enough to ensure connectivity. This requirement seriously hinders the use of the random key predistribution schemes when sensor networks are sparse. Second, the idea of key (or key space) sharing in these schemes, a requisite for high network connectivity, also degrades resilience against node capture.

Our Contribution

We propose a Communication Localized Group-based pairwise key predistribution scheme (CLG) to establish pairwise keys between each pair of neighboring sensors in a large sensor network. As in previous work [Du04, Huang04], we use the fact that typical sensor deployments are group-based, so that sensors deployed in the same group are more likely to be neighbors.

In our scheme, each sensor will be preloaded with unique pairwise keys shared with all other sensors in the same group. This is feasible, with a reasonable group size. Previous work based on group-based deployment model [Du04, Huang04] typically assumes that group adjacencies are known prior to sensor deployment. In contrast, our two deployment methods are more flexible and do not need that assumption. For every pair of neighboring sensors from different groups, we use a technique involving local communication for path key establishment. Each path key establishment involves at most two intermediate nodes. The uniqueness of pairwise keys enables the CLG scheme a graceful degradation of security as the number of compromised sensor increases, hence significantly improving the resilience against node compromise over the random key predistribution schemes.

Further, in CLG, the communication required for a path key establishment is localized to two adjacent groups. Therefore, as our analysis shows, CLG significantly reduces the communication overhead compared to PIKE [Chan05], which requires network-wide communication for path key establishment.

Our scheme has the following salient features:

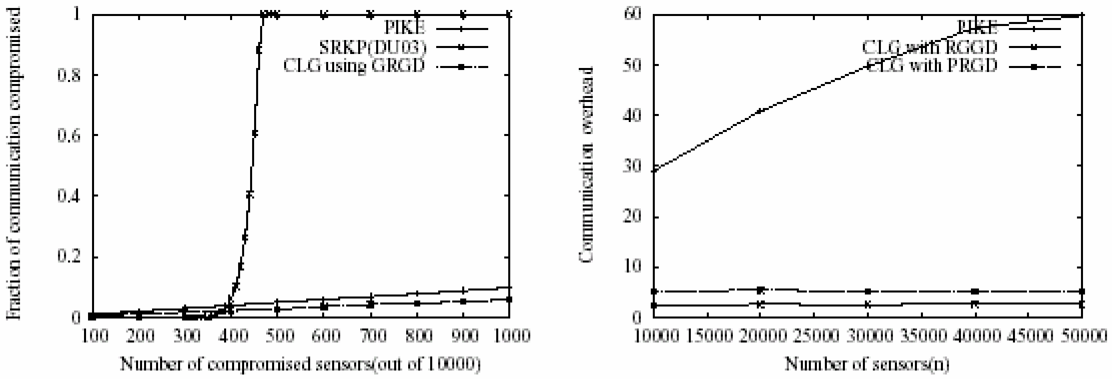
Density and distribution independence: CLG ensures the establishment of pairwise keys between any pair of neighboring sensors, regardless of sensor density or distribution. It is more general than the random key predistribution schemes, which require uniform sensor deployment with high density.

Graceful resilience degradation: CLG does not suffer dramatic degradation as the number of compromised sensors increases. Even with a large fraction of nodes compromised, only a small fraction of secure links are compromised in the rest of the sensor network. CLG provides stronger resilience against node capture than the random key predistribution schemes.

Localized communication: CLG is communication-free for each pair of neighboring sensors from the same groups to establish a pairwise key, and only involves local communication for each pair of neighboring sensors that are in different groups to establish a pairwise key.

Low memory requirements: CLG has low memory requirements. For a sensor network of *n* sensors, with group size n_G, our scheme requires each sensor to be preloaded with $\frac{1}{2}(n_G-1+(n-n_G)/n_G^2)$ keys.

Mobility support: CLG is suitable for mobile sensor networks.



(a) Resilience Comparison among CLG, PIKE and SRKP.

(b) Communication Overhead Comparison between CLG and PIKE.

Figure 22: Resilience and Communication Overhead Compared

8. Indexing Moving Objects: A cost model based approach

Problem Statement: Spatiotemporal queries answer regarding absolute or relative locations occupied by mobile objects during time intervals of interest. Our goad is to develop index structure and algorithms for such queries.

Motivation: Spatiotemporal queries are important for many current applications, such as air traffic control, transportation systems, and weather forecasting, as well as for emerging applications such as location-based services and digital battlefields. Another important application area is Mobile Ad-hoc Networks (MANETs), which are local area wireless networks, with wireless devices as mobile nodes.

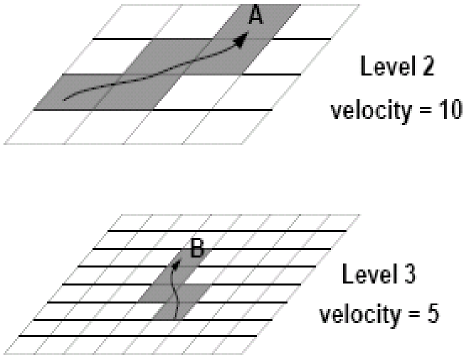


Figure 23: A sample vTree

Approach: Our approach, the vTree index, is to partition a dataset both by velocity as well as along spatial dimensions. We segregate objects by velocity, and separate the index components for different velocity ranges, so that uncertainties due to the high-velocity objects do not color locality information pertaining to low-velocity objects. A vTree is a hierarchical structure, and partitions space coarsely at its higher tiers and more finely at its lower tiers. To accommodate complex motions, an object's descriptors are strategically replicated in the grid cells overlapping its trajectory.

To minimize replication, we place objects with higher velocities in the upper tiers, and those with low velocities at lower tiers. As a result, location information about low-velocity objects is maintained at higher resolutions than high-velocity objects. This approach has two benefits. First, the information pertaining to low-velocity objects is not affected by the uncertainties due to high-velocity objects. Second, reducing replication reduces the space requirement.

A vTree also carefully exploits locality of access. Algorithms for processing vTrees are designed to limit the portion of the index and data space to be explored in responding to a query, as well as to maximize the degree of locality within the portion of space to be explored. Locality is exploited by using space filling Hilbert curves to order the items stored at a particular tier. We provide analytical estimates for join processing times in vTree. Our experiments demonstrate that the analytical model accurately captures the complexity of the join algorithm.

Experiments:

Figure 24 compares for range query with the approach presented in [2]. This figure shows the cost of join for datasets of varying sizes. The parameters of experiment were: IndexLifetime: 60 mins; query distance = 30 miles; query time = 25 mins.

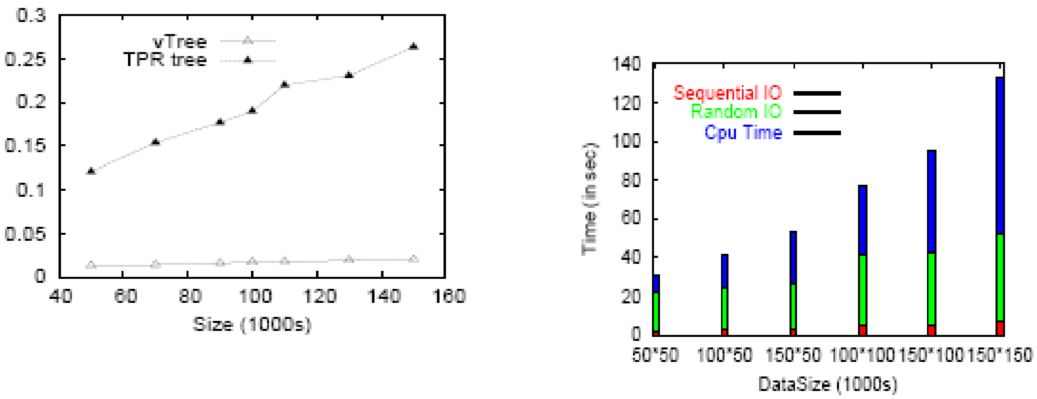


Figure 24: Join performance, varying dataset sizes.

9. Roads, Codes, & Spatiotemporal Queries

Problem Statement: Develop a theoretical framework for answering spatial and spatiotemporal queries on objects moving along a system of curves on the plane such as many planar road networks. Devise algorithms for join, range, intercept, and other spatial and spatiotemporal queries under these assumptions, with distances being measured along the trajectories.

Motivation: Surprisingly, most of existing work considers Cartesian (typically, Euclidean) spaces, where the distance between two objects is determined solely by their relative position in space. However, in practice, objects can usually move only on a pre-defined set of trajectories as specified by the underlying network (road, railway, river etc.). Thus, the important measure is the network distance, i.e., the length of the shortest trajectory connecting two objects, rather than Euclidean distance.

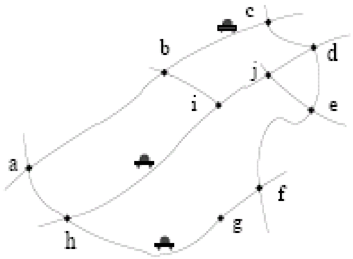


Figure 25: Planar Road Network

Approach: Central to our approach is an efficient coding technique, based on hypercube embedding, for assigning labels to nodes in the network. The Hamming distance between codes corresponds to the physical distance between nodes, so that we can determine shortest distances in the network extremely quickly. The coding method also efficiently captures many properties of the network relevant to spatial and spatiotemporal queries. Our approach also yields a very effective spatial hashing method for this domain

Contribution: Our analytical results demonstrate that our methods are space- and time-efficient. We have studied the performance of our method for large planar graphs designed to represent road networks. Experiments show that our methods are efficient and practical. Figure 26 shows the size of labels with respect to table lookup. Figure 27 compares time to compute shortest distance with Dijkstra.

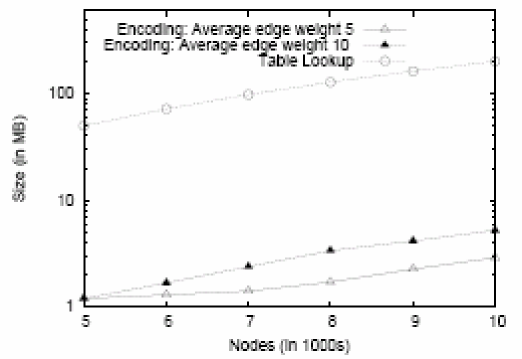


Figure 26: Table Lookup vs Encoding

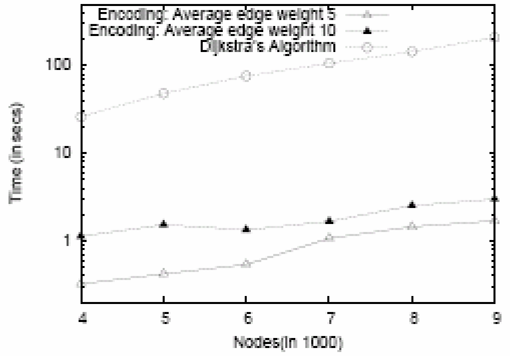


Figure 27: Time to compute Shortest Path

10. Efficient Data Dissemination Using Locale Covers

Problem Statement: Develop a scalable and efficient dissemination scheme for location-dependent data over resource constrained channel.

Motivation: Location-dependent data are central to many emerging applications, ranging from traffic information services to sensor networks.

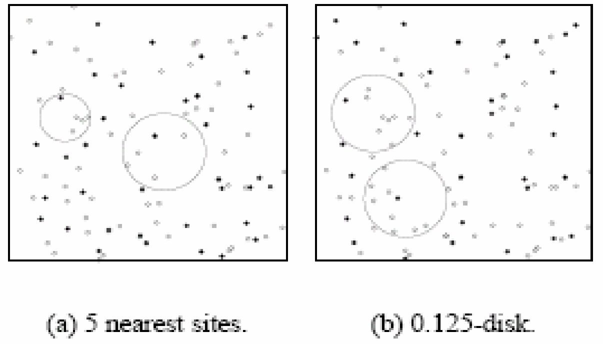


Figure 28: Example Locale Cover

Approach: The standard pull- and push-based data dissemination models become unworkable since the data volumes and number of clients is high. We address this problem using locale covers, a subset of the original set of locations of interest, chosen to include at least one location in a suitably defined neighborhood of any client. Since location-dependent values are highly correlated with location, a query can be answered using a location sufficiently close to the query point. Typical closeness measures might be Euclidean distance, shortest distance on graphs, or a k-nearest neighbor criterion.

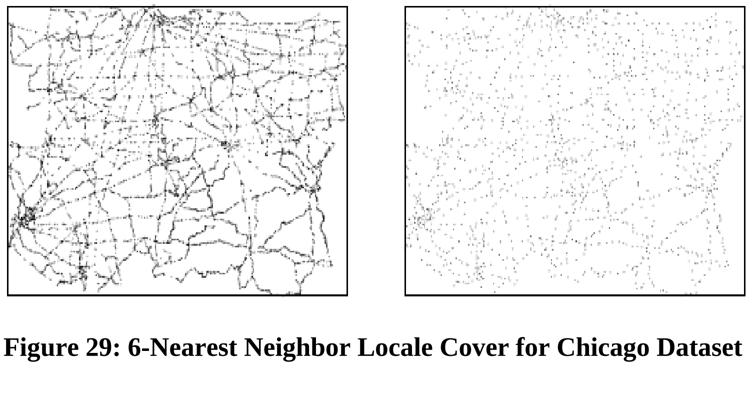


Figure 29: 6-Nearest Neighbor Locale Cover for Chicago Dataset

We also introduce a nested locale cover scheme that ensures fair access latencies, and allows clients to refine the accuracy of their information over time.

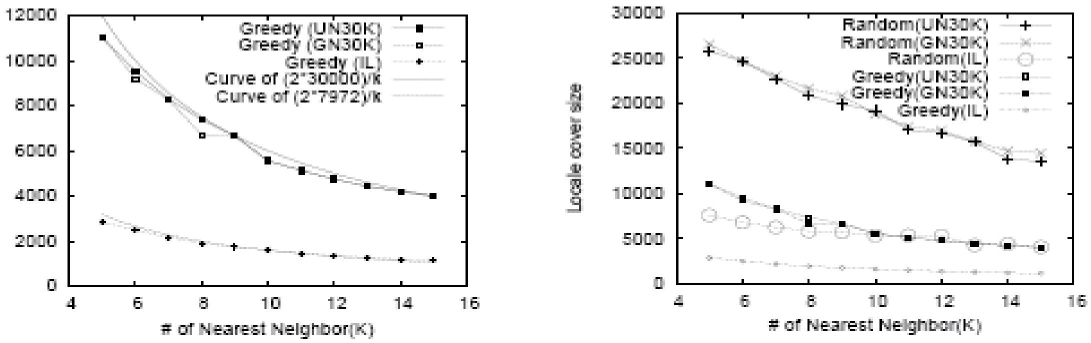


Figure 30: Greedy vs Random Sampling

Results: We show that location-dependent queries may be answered satisfactorily using locale covers, with small loss of accuracy. Our approach is independent of locations and speeds of clients, and is applicable to mobile clients. We also prove two important results: One regarding the greedy algorithm for sensor covers and other pertaining to randomized locale covers for k-nearest neighbor queries. Figure 30 shows the sizes of k-domain locale cover via greedy and random sampling approach for 30,000 points with uniform and skewed distribution respectively, and compares greedy and random approach.

11. Internet Key Service (IKS)

Problem Description

In the modern Internet the administrative burden of naming entities is decentralized via the hierarchical DNS namespace. The distribution of top-level domain names is strictly controlled by the IETF and the administration of these top-level domains is handled by a handful of top-level domain name servers administered by a variety of organizations. These globally unique DNS names are integrated into host names, email addresses, URL's and numerous other global Internet identifiers. DNS decentralizes the administration of DNS names by allowing an organization possessing authority over a domain to delegate authority for administering a sub-domain to a different organization. For example, the University of California, Riverside, possessing authority over the domain *ucr.edu*, delegates authority for the domain *cs.ucr.edu* to the Computer Science & Engineering Department, which administers a DNS server for *cs.ucr.edu*.

Currently the Domain Name System is not secure; name resolution responses are not authenticated by the authoritative name servers for domains. This deficiency allows DNS server responses to be impersonated, corrupting this crucial Internet service. In order to secure DNS the IETF has sponsored the DNSSEC working group to develop DNS security extensions to cryptographically authenticate responses from DNS. The DNSSEC proposal is quickly approaching deployable status. It is anticipated that 2005 may see the first steps towards widespread DNSSEC deployment.

In order to bypass the present insecurity of DNS current public key infrastructures utilize an orthogonal set of root Certificate Authorities (CAs), which are maintained by many of the same organizations responsible for administering the root DNS name servers. These root CAs are responsible for generating certificates vouching for the identity of servers utilizing public key cryptography to secure the services they implement. Since these identity certificates are only loosely tied to the DNS namespace the verification necessary to administer this security infrastructure is substantial and severely limits the ability to use delegation to form a certificate chain.

With the coming deployment of DNSSEC it will no longer be necessary, or advantageous, to maintain this separate security infrastructure. DNS ultimately controls the delegation of domain names for the services and servers utilizing these names. With authenticated DNS responses it becomes desirable to allow public keys registered to DNS-name based identifiers to be directly bound to the DNS hierarchy. A necessary requirement of such a system is that it must not place a substantial burden on the DNS, since harming this crucial Internet infrastructure would be disastrous.

11.1 Our Contribution

We have developed the Internet Key Service (IKS), a new public key infrastructure that successfully leverages the strong authentication provided by DNSSEC to allow Internet services and servers to securely publish public keys. IKS achieves this objective without placing a substantial burden on DNS / DNSSEC. By tightly integrating the IKS public key infrastructure with DNS we minimize the administrative burden of publishing per-entity keys.

Since DNS controls the binding of DNS names to host addresses we feel it is preferred to use DNS to also provide the foundation for the binding of DNS names to public keys held by the entities identified by these names. Since the DNS server for *cs.ucr.edu* has the sole authority of assigning the address of the host *www.cs.ucr.edu* we feel it is most appropriate for clients wishing to discover an authenticated public key for *www.cs.ucr.edu* to utilize a service that obtains its authority to bind public keys to *www.cs.ucr.edu* directly from the DNS server for *cs.ucr.edu*. This property is a central feature of IKS. Without the ability to authenticate DNS responses, provided by DNSSEC, it is impossible to provide this coupling.

With IKS a domain administrator may provide CA services to entities with DNS names belonging to this domain by publishing the identity of an IKS server that is responsible for maintaining bindings between DNS named entities and the public keys they use to secure their services. Domains choosing not to publish the identity of an IKS server simply do not provide CA services to entities belonging to this domain. Of particular importance top-level domains need not provide IKS servers since the principal components of these domains are sub-domains rather than named entities, for example no one uses email addresses such as *user@com*.

A client wishing to discover a public key for a service associated with a specific DNS named entity begins by querying the DNSSEC enabled DNS server for the entity's domain for the identity of the authoritative IKS server. The identity of the IKS server for a domain is published as a DNS SRV record. The client can then query this IKS server for a listing of all the published public keys associated with the desired entity, along with structured metadata describing the format and usage of each public key, including the service it is registered for. The format of this response is a signed XML datagram. The client can authenticate this list of published keys using a fingerprint of the signing key, which is published as a DNS TXT record in the DNSSEC enabled DNS server. After validating the authenticity of the retrieved list of published keys the client can use the included metadata to locate the key identifier for a published public key registered for the desired service and meeting the required security parameters. The client can then use this key identifier to fetch the public key from the IKS server.

DNS named entities may register a public key for a specific service by submitting an authenticated registration request to the IKS server. Upon authenticating this request the IKS server updates this entity’s list of published keys to include this key. Entities may revoke previously published keys by submitting a similarly structured key revocation request to the IKS server.

Key registration and revocation requests must be authenticated by the IKS server to ensure that the requesting individual is authorized to register and/or revoke keys for the specified entity. In order to provide a simple yet flexible authentication system we allow three methods of authenticating these requests. The simplest authentication mechanism we support is to authenticate requests by including an encrypted username/password that can be verified against the entity. Alternatively the request can be signed by a key registered as a ‘Key management key’ by the entity. Finally we provide a general mechanism by which a trusted third-party authentication server can authenticate the request and attach an authentication signature to the request.

Since IKS requests are automatically distributed by DNS domains, only large domains, containing a great many hosts and/or users face difficulties with IKS. These large domains such as *aol.com* may improve their scalability by invisibly partitioning the domain by hashing the entity names into a pool of IKS servers. This horizontal partitioning of the IKS server eliminates interdependencies and allows a large domain to be treated just as if it were a collection of smaller domains.

11.1.1 RIKS: An IKS Prototype

In order to further our understanding of the deployment issues of IKS we have constructed a prototype IKS implementation, the Riverside Internet Key Server (RIKS). RIKS was written as an Apache Mod_Python web service. In addition to the RIKS IKS server prototype we have developed a Python based IKS client library, which handles the details of discovering the IKS server from the entity's DNS domain server, looking up and authenticating registered keys.

RIKS is implemented as three independent processes, a query process that handles clients' key lookup queries, a registration/revocation process that handles entities' key registration and revocation requests and a signature generation process which periodically generates pre-signed XML responses to queries made through the query process, based upon recent modifications made by the registration/revocation process. These three components communicate through a SQL database, which serves as a backing store for the published key database. The primary reason for this separation is to insulate the sensitive signing keys being used by the signature generation process from client requests. Additionally this separation allows separate machines to handle each of these specialized workloads.

Our experience with the RIKS prototype validates our belief that IKS can be efficiently implemented and deployed to provide an authenticated key distribution system for DNS named entities without placing a substantial burden on the DNS infrastructure.

12 Denial of Service Attacks at the MAC layer in MANETs

A fault-tolerant and reliable ad hoc network has to be able to withstand Denial of Service (DoS). In such attacks, a flood of dummy messages blocks the functionality of the network. While DoS attacks have been studied extensively for the wire-line networks, there is lack of research for preventing such attacks in ad hoc networks. Considering their deployment in tactical battlefield missions, these networks are likely to be attacked by malicious intruders. In this work, we study DoS attacks and propose solutions to alleviate them. In more detail, we first try to define the enemy: we identify the weaknesses of the current network architecture, which an intelligent attacker would exploit. In an ad hoc network, an attacker has significantly different capabilities and restrictions than those of a wireline attacker. Finally, we develop a suit of protocols that can alleviate the attacks that we identify. Our work is based in introducing fairness both at the MAC and routing layer. Namely, we use local and connection-level fairness. Our work is among the first attempts to safeguard ad hoc networks.

This project consists of three phases:

1. What would an intelligent attack be like? We develop intelligent attack strategies and identify the features that make the network vulnerable and quantify the effects of DoS. We find that MAC layer unfairness is the main weakness of the network.
2. We examine whether a fair MAC layer would be sufficient for suppressing DoS attacks. We find that, while MAC layer fairness is necessary, it is not sufficient. We show that fairness at the MAC layer when enhanced by network layer fairness is able to prevent DoS attacks.
3. We develop a suit of strategies to alleviate DoS and reduce their impact. Our approach is tunable to address the different levels of security required. In our approach, we control a carefully chosen set of parameters such as the continuous occupation of medium by a transmitting station, frame size and packet retransmissions counter. In addition, our approach improves the throughput in the network.

We are currently fine-tuning the third and final phase of the project building on the initial successes of our approach.

Phase I. We show that Transmission Control Protocol (TCP) based services can be obliterated by intelligent DoS attacks. The primary weakness that is exploited by the attacker is the unfairness and the related capture phenomenon of IEEE 802.11 protocol. These 802.11 weaknesses are exploitable because of unsophisticated design of the network layer. Thus, starting a heavily loaded UDP based flow in the vicinity of TCP flows can launch these attacks. An attack that congests the medium locally, thus preventing access for other nodes is termed as **jamming attack** as shown in Figure 31. An attack that creates congestion in a neighborhood by routing packets through it is called **routing attack**.

We discuss some interesting DoS attacks in the wireless environment and suggest possible solutions. Our description is a brief and exhaustive listing of such attacks; their prevention methods, and the evaluation of prevention methods are beyond the scope of this paper. In wireless networks DoS attacks could be mainly classified into two types, those that are at the routing layer and those that are at the MAC layer. Attacks at the routing layer could consist of the following:

a) The malicious node participates in a route but simply drops a certain number of the data packets. This causes the quality of the connections to deteriorate and further ramifications on the performance if TCP is the transport layer protocol that is used.

b) The malicious node transmits falsified route updates. The effects could lead to frequent route failures thereby deteriorating performance.

c) The malicious node could potentially replay stale updates. This might again lead to false routes and degradation in performance.

d) Reduce the TTL (time-to-live) field in the IP header so that the packet never reaches the destination.

Notice that all of the above could lead to congestion due to data that is either retransmitted or transmitted on erroneous routes only to be dropped at a later time. Some of these issues are addressed in recent literature One method proposes the use of the promiscuous mode wherein a node overhears the transmission of its neighbors and infers if the behavior and responses are normal. However, this overhearing may be very much dependent upon other transmissions in the vicinity and the MAC protocol in use. It has also been proved that if end-to-end authentication is enforced, attacks by independent malicious nodes of types (b) and (c) may be thwarted. An attack of type (a) may be handled by assigning confidence levels to nodes, and using routes that provide the highest level of confidence. Of course, multiple paths might have to be maintained. An attack of type (d) may be thwarted simply by making it mandatory that a relay node ensures that the TTL field is set to a value greater than the hop count to the intended destination. If nodes collude, the authentication mechanisms fail and it is an open problem to provide protection against such routing attacks.

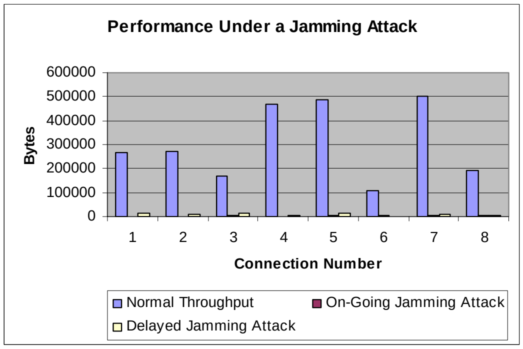


Figure 31: DOS attack in ad-hoc network. TCP throughput is negligible.

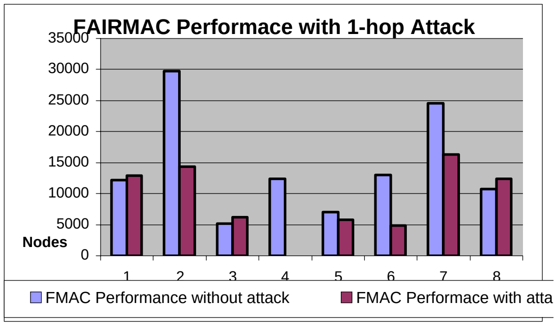


Figure 25: Use of FAIRMAC protects throughput during attack.

At the MAC layer the following attacks can be attempted:

a) Since we assume that there is a single channel that is re-used, keeping the channel busy in the vicinity of a node leads to a denial of service attack at that node.

b) By using a particular node to continually relay spurious data the battery life of that node may be drained.

An end-to-end authentication may prevent these attacks from being launched. If the node does not include a certificate of authentication it might be prevented from accessing the channel. However, if nodes collude and one of the nodes is the sending node and the other is the destination, MAC layer attacks are very much feasible.

The primary results of phase 1 were the following:

- Unfairness at the MAC layer is the main weaknesses that can lead to a DoS.
- It is possible to launch DoS attacks on a TCP based service with both a jamming and a routing attack.

The attacks cannot be thwarted by simple localized measures like stabilizing routing or increasing MAC layer robustness.

Phase II. Motivated from the results of Phase I, we wanted to see if having a fair MAC protocol would improve the performance and the resistance to DoS attacks. For this, we consider a fair MAC protocol, which we call FAIRMAC. Note that FAIRMAC is not intended to be a deployable protocol but a mechanism to explore the effect of the MAC on the attack. Through extensive simulations, we showed that:

- Jamming attacks can be completely eliminated by MAC layer fairness as shown in Figure 32.
- Routing attacks cannot be eliminated by MAC layer fairness.

Next, we enhanced the fairness at the MAC layer by implementing various network layer fair-queuing mechanisms. In particular, we showed that

- Locally fair queuing mechanisms fail to prevent DoS even with a fair MAC.
- IP-destination based queuing provides greatest robustness against DoS attacks

The observations of this phase provided the insight that was required for Phase III.

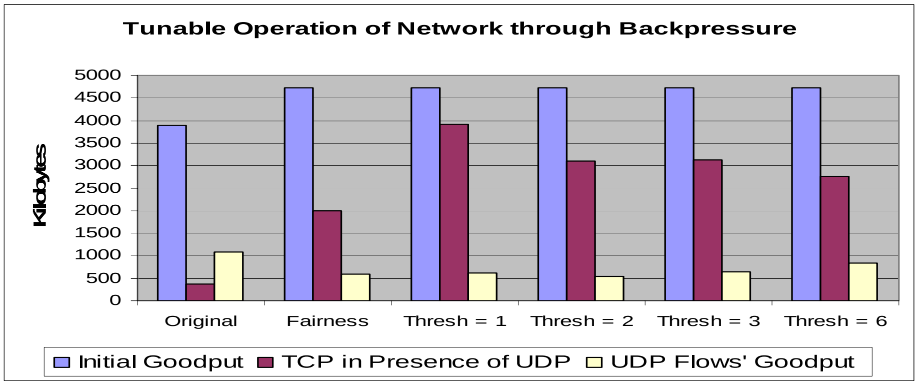


Figure 263: Tunable protection against aggressive UDP flows.

Phase III. In this phase, we developed a scheme that prevents UDP based flows from harming the TCP based services. The scheme does not encumber the UDP based flows, rather provides tunable operation of the network. The fundamental parameter of interest is the continuous occupation of medium by a transmitting station. Tuning this parameter along with the frame size, and retry counter limits used at the MAC layer we are able to offer a wide choice in network operation, varying from a UDP supportive insecure system to TCP supportive secure system. It should be noted that our solution improves system performance besides securing it. Sample results of the scheme are shown below.

Backpressure: resisting DoS attacks. We propose **backpressure**, a scheme that significantly improves TCP performance in the presence of UDP flows.   In a nutshell, our scheme provides burst regulation at the flow level at each node. A forwarding node refuses to accumulate large number of packets from a flow. Once the buffering quota is reached, the node will not accept more packets from that flow before it can forward some of the buffered packets. We show that our scheme can provide a coarse control over bandwidth allocation to TCP and UDP streams. Finally, an advantage of our scheme is that it can be implemented on top of the IEEE 802.11 MAC and does not require any changes to TCP.

To implement back-pressure, we set a threshold, referred to as the **back-pressure threshold** that restricts the buffer allocation to a particular IP-source (or IP-destination) at any node. Then, using the promiscuous mode of operation, a node keeps track of the number of packets in a downstream neighbor's queue. Upon receiving a MAC frame, the nodes operating in a promiscuous mode can determine if a neighbor has transmitted a packet belonging to a particular IP-source (or destination). Thus, for a flow, each upstream node is aware of the queue size at a downstream neighbor to which it forwards packets from the flow. Once the node recognizes that the back-pressure-threshold is reached at a downstream neighbor, it stops transmitting packets from the flow to that neighbor. Subsequently, the node's back-pressure limit is reached which would prevent its previous upstream relay from sending further packets from that flow. This effect is propagated all the way to the source of the flow.

In Figure 33, we plot the packets sent and received by UDP and TCP agents for IP-Source fair-queuing and UDP back-pressure. Notice that with back-pressure, the number of packets injected into the network by the UDP source is almost one-fifth (22%) of that with the simple IP-source fair-queuing. However, the number of packets actually delivered by the UDP flows is almost equal in both cases (back-pressure in fact results in the delivery of 4.5% more packets). With the increased medium availability TCP clients achieve a better throughput. This rate adaptation of the TCP source also leads to a reduction in MAC and Interface queue (IFQ) related drops for the TCP connections as well as for UDP flows.

Tuning the backpressure threshold, we are able to offer variable TCP and UDP goodput. For example, tuning the threshold value to 1, even under attack the TCP clients achieve 82% of the goodput under normal conditions. Tuning to threshold values of 3 provides greater UDP throughput at the cost of TCP goodput. In other words, larger threshold enables bursty flows, at the expense of well behaved TCP flows.

Another significant benefit of backpressure is that it reduces the jitter in the end-to-end delay of the UDP flows' packets. Further, we also witness an increase in jitter with increase in backpressure threshold value.

This work has lead to two conferences [Gupta02, Gupta04] and is being submitted to a journal. More details on this project can be found there.

References

[Ayanoglu93] E. Ayanoglu, I. Chih-Lin, R.D. Gitlin, and J.E. Mazo, “Diversity coding for transparent self-healing and fault-tolerant communication networks,” IEEE Transactions on Communications, vol. 41, no. 11, pp. 1677–1686, 1993.

[BYS04] Berger D., Ye, Z., Sinha P, Krishnamurthy S.V., Faloutsos M., and Tripathi S.K., “TCP Friendly Medium Access Control for Ad Hoc Networks: Alleviating Self-Contention”, IEEE MASS 2004, Ft. Lauderdale.

[Chen02] X. Chen, M. Faloutsos, and S.V. Krishnamurthy, “Distance Adaptive (DAD) Broadcasting for Ad Hoc Networks”, in Proceedings of MILCOM 2002.

[Chen03] X. Chen, M. Faloutsos, and S.V. Krishnamurthy, “Power Adaptive Broadcasting with Local Information in Ad Hoc Networks”, in Proceedings of IEEE ICNP 2003.

[Eriksson03] J. Eriksson, M.Faloutsos, S. Krishnamurthy, “PeerNet: Pushing Peer-to-Peer Down the Stack”, IPTPS, 2003.

[Ge04] M. Ge, S.V. Krishnamurthy, and M. Faloutsos, “Overlay Multicasting in Ad Hoc Networks”, in Proceedings of Med-Hoc-Net, 2004.

[GR04a] Sandeep Gupta, Swastik Kopparty, Chinya Ravishankar, “Roads Codes and Spatiotemporal Queries”, Proceedings of the Twenty-third ACM Symposium on Principles of Database Systems (PODS), June 14-16, 2004, Paris, France.

[GR04b] Sandeep Gupta, Chinya Ravishankar, “Using vTree Indices for Queries over Objects with Complex Motions”, Proceedings of 20th International Conference on Data Engineering, ICDE 2004.

[GR05]Sandeep Gupta, Jinfeng Ni, Chinya Ravishankar, “Data Dissemination Using Locale Covers”, Under Submission, Very Large Data Bases (VLDB), 2005.

[Gui03] C. Gui and P. Mohapatra, “Efficient Overlay Multicast for Mobile Ad Hoc Networks”, in Proceedings of IEEE WCNC 2003.

[Gupta02] V. Gupta, S.V. Krishnamurthy and M. Faloutsos, "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks", in Proceedings of MILCOM - Network Security, Anaheim, October 2002.

[Gupta04] V. Gupta, S.V. Krishnamurthy and M. Faloutsos, “Improving the Performance of TCP in the Presence of Interacting UDP Flows in Ad Hoc Networks”, in Proceedings of IFIP Networking 2004, Athens, Greece.

[Hong99] X. Hong, M. Gerla, G. Pei, and C. Chiang, “A group mobility model for ad hoc wireless networks,” 1999.

[HV99] G. Holland and N. Vaidya, “Analysis of TCP Performance over Mobile Ad Hoc Networks”, Proceedings of ACM MOBICOM 1999.

[Jeff00] J.E. Wieselthier, G. D. Nguyen and A. Ephremides. “On construction of energy-efficient broadcast and multicast trees in wireless networks”, in Proceedings of IEEE INFOCOM, 2000.

[JLK05] Jakllari, G., Luo, W., and Krishnamurthy, S.V., “A Polling based MAC protocol for use with directional antennae in mobile ad hoc networks”, submitted to IEEE WOWMOM, Taormina, 2005.

[KKF02] S. Kopparty, S.V.Krishnamurthy, M.Faloutsos and S.K.Tripathi, “Split-TCP for Ad Hoc Networks”, Proceedings of IEEE GLOBECOM 2002.

[KKT03] F.Klemm, S.V. Krishnamurthy and S.K.Tripathi, ``Alleviating Effects of Mobility on TCP Performance in Ad Hoc Networks using Signal Strength based Link Management”, Proceedings of IFIP Personal and Wireless Communications (PWC), 2003, Venice, Italy.

[KYK05] F.Klemm, Z.Ye, S.V. Krishnamurthy and S. K.Tripathi, ``Improving TCP Performance in Ad Hoc Networks Using Signal Strength Based Link Management”, The Ad Hoc Networks Journal, March 2005.

[Law04] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, “On Evaluating the Trade-offs between Broadcasting and Multicasting in Ad Hoc Networks”, in Proceedings of IEEE MILCOM 2004.

[Lee00] S.J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols”, in Proceedings of IEEE INFOCOM, 2000.

[Liang02] W. Liang, “Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks”, in Proceedings of ACM MOBIHOC, 2002.

[Marina01] M.K. Marina and S.R. Das, “On-demand multipath distance vector routing in ad hoc networks,” Proceedings of the International Conference for Network Procotols (ICNP), pp. 14–23, Nov. 2001.

[Maxemchuk75] N.F. Maxemchuk, “Dispersity routing in store and forward networks,” Ph.D. thesis, University of Pennsylvania, May 1975.

[Motwani95] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[Nasipuri01] A. Nasipuri, R.Castaneda, and S.R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” ACM/Kluwer Mobile Networks and Applications (MONET), vol. 6, no. 4, pp. 339–349, 2001.

[Pearlman00] M.R. Pearlman, Z.J. Haas, P. Sholander, and S.S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks,” Proceedings of the ACM MobiHoc, pp. 3–10, 2000.

[Pei00] G. Pei, M. Gerla, and X. Hong, “Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility,” in ACM MobiHOC’00, 2000.

[Pei99] Guangyu Pei, Mario Gerla, Xiaoyan Hong, and Ching-Chuan Chiang, “A wireless hierarchical routing protocol with group mobility,” in WCNC, 1999.

[Peng00] W. Peng and X.-C. Lu, “On the reduction of broadcast redundancy in mobile ad hoc networks”, in Proceedings of the ACM MOBICOM, 2000.

[PR99] C.E. Perkins, and E. M. Royer, “Ad-hoc On-Demand Distance Vector Routing”, Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, February 1999.

[Ramanathan98] Ram Ramanathan and Martha Steenstrup, “Hierarchically-organized, multihop mobile wireless networks for quality-of-service support,” Mobile Networks and Applications, vol. 3, no. 1, pp. 101–119, 1998.

[RR02] R.Roychoudhury et al, ``Using Directional Antennas for Medium Access Control in Ad Hoc Networks'', in Proceedings of ACM MOBICOM, 2002.

[SJLL00] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A.Lopez. Indexing the positions of continuously moving objects. In *SIGMOD Conference*, pages 331–342, 2000.

[SKP04] Shah, V, Krishnamurthy S.V., and Poojary, N., “Improving MAC Layer Performance in Ad Hoc Networks with Nodes with Heterogeneous Transmit Power Capabilities”, IEEE ICC 2004, Paris.

[SX02] T.Saadawi and S.Xu, “Performance Evaluation of TCP Algorithms in Multi-hop Wireless Packet Networks”, Journal of Wireless Communications and Mobile Computing 2002.

[Wu01] K. Wu and J. Harms, “Performance study of a multipath routing method for wireless mobile ad hoc networks,” Proceedings of the IEEE Int’l Symposium on Modeling, Analysis and Simulation of Compute and Telecommunication Systems (MASCOTS), pp. 99–107, 2001.

[YBS03] Ye, Z., Berger D., Sinha P., Krishnamurthy S.V., Faloutsos M., and Tripathi S.K., “Alleviating MAC Layer Self-Contention in Ad Hoc Networks”, Poster at ACM MOBICOM 2003, San Diego.

[Ye03] Zhenqiang Ye, Srikanth V. Krishnamurthy, Satish K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks". INFOCOM 2003, San Francisco.

[YKT04a] Z.Ye, S.V. Krishnamurthy and S.K.Tripathi, “Effects of Multipath Routing on TCP Performance in Ad Hoc Networks”, IEEE GLOBECOM 2004, Dallas.

[YKT04b] Z.Ye, S.V. Krishnamurthy, and S. K. Tripathi, “Use of Congestion Aware Routing to Spatially Separate TCP Connections in Wireless Ad Hoc Networks”, IEEE MASS 2004, Ft. Lauderdale.